Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Samo Kralj

Poravnava zvočnih in notnih zapisov ljudske glasbe

magistrsko delo

študijski program druge stopnje
računalništvo in informatika

Mentor: doc. dr. Matija Marolt

Ljubljana, 2016

Izjava o avtorstvu magistrskega dela

Spodaj podpisani Samo Kralj sem avtor magistrskega dela z naslovom:

Poravnava zvočnih in notnih zapisov ljudske glasbe

S svojim podpisom zagotavljam, da:

- sem magistrsko delo izdelal samostojno pod mentorstvom doc. dr. Matije Marolta,
- so elektronska oblika magistrskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko magistrskega dela,
- soglašam z javno objavo elektronske oblike magistrskega dela v zbirki "Dela FRI".

V Ljubljani, 20. julij 2016 Podpis avtorja:

Zahvaljujem se svojemu mentorju doc. dr. Matiji Maroltu za pomoč in vodenje pri izdelavi magistrske naloge.

Posebna zahvala velja še staršema, ki sta me podpirala pri študiju.

Ljubim tistega, ki sanja o nemogočem.
(Johann Wolfgang von Goethe)

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
QBH	query by humming or singing	poizvedba s pomočjo žvižganja ali petja
FS	frequency spectrum	frekvenčni spekter
MIDI	musical instrument digital interface	
XML	extensible markup language	
WAV	waveform audio file format	
DP	dynamic programming	dinamično programiranje
DTW	dynamic time warping	dinamično časovno krivljenje
MP	Markov process	Markov proces
HMM	hidden Markov model	skriti model Markova
MIR	music information retrieval	rudarjenje po glasbenih podatkih
MIREX	music information retrieval evaluation exchange	

Povzetek

V magistrski nalogi obravnavamo problem poravnave zvočnih in notnih zapisov za ljudsko glasbo. Za cilj si zadamo prilagoditi standardne metode za poravnavo zvočnih in notnih zapisov za uporabo na arhivu ljudske glasbe. Algoritme za poravnavo zvočnih in notnih zapisov nato uporabimo za iskanje v arhivu ljudske glasbe. Pri ljudski glasbi se pogosto srečamo s problemom slabše kakovosti petja pevcev, ki pogosto niso profesionalni pevci. Slabša kakovost zvočnih zapisov oteži iskanje po arhivu glasbe. V magistrski nalogi preizkusimo nekaj pristopov za izboljšanje učinkovitosti iskanja po arhivu ljudske glasbe. Pri načrtovanju rešitev upoštevamo lastnosti ljudske glasbe kot so specifična porazdelitev tonov pri ljudski glasbi. Naše rešitve preizkusimo na arhivu slovenske ljudske glasbe, kjer dosežemo pomembno izboljšanje rezultatov v primerjavi s standardnimi metodami.

Ključne besede: ljudska glasba, poravnava glasbenih zapisov, poizvedba s petjem, dinamično časovno krivljenje, skriti model Markova

Abstract

The thesis deals with the problem of audio to score alignment of folk music. Our goal is to adapt standard methods for audio to score alignment for aligning folk music. Alignment algorithms are then used for querying music in folk music archives. In folk music, the quality of recordings is often poor, due to recording conditions, and because singers are often not professionals. This makes alignment with standard approaches difficult. In our master thesis we develop new approaches to improve alignment in folk music archives by exploiting the characteristics of folk music, such as its tone distribution. We test our algorithms on slovenian folk music and show that we achieve improvement over standard methods.

Keywords: folk music, music alignment, query by humming, dynamic time warping, hidden Markov model

Poglavje 1

Uvod

V zadnjem času smo priča izjemnemu porastu količine podatkov, ki jih hranimo na različnih medijih. Da bi te podatke lahko koristno uporabili, moramo zagotoviti učinkovito iskanje po gruči podatkov ter učinkovito ekstrakcijo informacij. V tem magistrskem delu se ukvarjamo z glasbenimi zapisi, oziroma še bolj ozko z glasbenimi zapisi ljudske glasbe. Te so v obliki različnih zvočnih in notnih zapisov. Naš cilj je zagotoviti učinkovito in robustno iskanje po arhivu ljudske glasbe. Da bi dosegli želeno, je potrebno najprej preučiti lastnosti glasbe ter glasbenih zapisov ter na podlagi le teh konstruirati primerne metode. V preteklosti je bilo največ raziskovalnega napora osredotočenega na področje klasične glasbe[1, 2]. Na področju ljudske glasbe je bilo teh raziskav manj, zato je cilj te naloge raziskati zakonitosti ljudske glasbe ter na podlagi teh konstruirati učinkovite metode za iskanje po arhivu ljudske glasbe.

Na področju obdelave zapisov glasbe se je izoblikovalo več različnih nalog, kot so iskanje po arhivu glasbe, anotacija zapisov glasbe, primerjava različnih izvedb istega glasbenega dela. Pri vseh pa je najprej potrebno zagotoviti poravnavo dveh glasbenih zapisov. Za poravnavo dveh glasbenih zapisov sta v tem kontekstu največkrat uporabljeni metodi dinamičnega časovnega krivljenja[3] in v zadnjem času vse večkrat tudi metoda skritega

modela Markova (HMM)[4]. Slednja je bolj učinkovita za poravnavo v realnem času[5, 6], to je v sistemih, kjer sledimo glasbenemu izvajalcu v realnem času ter na podlagi sledenja predvajamo ustrezno umetno spremljavo. Za nesprotno poravnavo pa je bolj učinkovita metoda dinamičnega časovnega krivljenja (DTW) [7, 8].

Na področju iskanja po arhivu glasbe je veliko pozornosti namenjene nalogi iskanja s pomočjo petja ali žvižganja (*query by humming*). To je tudi ena izmed standardnih nalog vsakoletnega tekmovanja MIREX, namenjenega ovrednotenju različnih sistemov MIR in algoritmov. Na področju iskanja s pomočjo petja ali žvižganja je pomembno delo [9]. V njem avtorji predstavijo princip predstavitve melodije s pomočjo treh simbolov, in sicer navzgor, enako, navzdol. V to obliko transformirajo tako petje ozr. zvižganje kot tudi originalno melodijo. Prednost te metode je njena hitrost, slabost pa relativna nenatančnost, kar je posledica neupoštevanja pomembnih informacij o glasbi, kot je višina tonov.

Rainer v članku [10] predlaga metodo z uporabo EMD (earth mover's distance) razdalje, ki je bila originalno namenjena razdaljam v transportu. Ta metoda uporabi informacije o višini in dolžini not v peti poizvedbi in ne primerja posameznih rezin skladbe s peto ali žvižgano poizvedbo. Na ta način je tudi ta metoda zelo hitra. Zaradi njene hitrosti se metoda večinoma uporablja v začetni fazi sistemov QBH.

Metoda DTW predstavlja naslednjo možnost za sisteme QBH [11]. Ker ta metoda uporabi informacijo o višini in času, doseže zelo dobro natančnost rezultatov. Njena slabost je relativna počasnost, ker primerja posamezne rezine posnetkov. Metoda se večinoma uporablja v zadnji fazi sistemov QBH.

Naslednja metoda, ki se uporablja v sistemih QBH, je metoda HMM [12]. Ta metoda poravna note na osnovi verjetnostnega modela za posamezne note ter v modelu upošteva še možnosti napak. V članku [13] so na podlagi testov ugotovili, da metoda HMM doseže večjo natančnost od metode DTW na zelo dobro zapetih poizvedbah, obraten rezultat pa je bil ugotovljen na množici

slabše zapetih poizvedb. Tudi metoda HMM je tako kot metoda DTW v primerjavi s preostalimi metodami QBH relativno počasna, doseže pa večjo natančnost.

Na področju poravnav zapisov ljudske glasbe predstavljajo pomembno osnovo dela Meinarda Mullerja[14, 15]. Za naše delo sta pomembna predvsem dva koncepta. Prvi je koncept kromagramov, to je predstavitev glasbenega zapisa, kjer frekvenčni spekter transformiramo v dvanajst dimenzij, za katere je podlaga dvanajst osnovnih tonov. Drugi pomemben koncept pa se nanaša na algoritem dinamičnega časovnega krivljenja, kjer z upoštevanjem periodične narave ljudskih pesmi prilagodimo implementacijo algoritma dinamičnega časovnega krivljenja na način, ki pomembno zmanjša časovno zahtevnost algoritma.

V magistrskem delu na podlagi analize zapisov ljudske glasbe izpopolnimo standardne metode za poravnavo glasbenih zapisov. Eden glavnih problemov, s katerim se srečamo, je slabša kvaliteta točnosti petja, intonacije na zvočnih zapisih. Ker v jedru standardnih metod leži primerjava glasbenih zapisov na podlagi značilk, ki jih dobimo iz frekvenčnega spektra, je potrebno iznajti metode, ki bodo robustne na omenjene napake. Za rešitev tega problema analiziramo različne metode za transformacijo netočnih glasbenih zapisov v bolj robustno obliko ter implementiramo robustne metode za računanje podobnosti med dvema časovnima rezinama posameznih sekvenc. Uspešnost metod preizkusimo na arhivu slovenske ljudske glasbe.

V naslednjem poglavju najprej predstavimo različne vrste glasbenih zapisov, kot so zvočni, notni, MIDI in MusicXML zapisi glasbe. V tretjem poglavju predstavimo različne zvočne značilke, s pomočjo katerih lahko različne glasbene zapise prevedemo v skupno predstavitev. Predstavitev različnih glasbenih zapisov v enaki obliki nam omogoči uporabo algoritmov za poravnavo glasbenih zapisov. Algoritmi za poravnavo glasbenih zapisov so

predstavljeni v četrtem poglavju, kjer se osredotočimo na dva najpomembnejša algoritma, ki se uporabljata za poravnavo glasbenih zapisov, to sta dinamično časovno krivljenje in skriti model Markova. V petem poglavju analiziramo značilnosti ljudskih glasbenih zapisov, da bi lahko predstavljene metode za iskanje po arhivih glasbe izpopolnili za iskanje po arhivu ljudske glasbe. V šestem poglavju nato opišemo implementacijo osnovnega algoritma za iskanje po arhivu glasbe ter predstavimo nekatere izboljšave. Med drugim predstavimo metode, ki upoštevajo verjetnostno porazdelitev tonov pri ljudskih pesmih, in metode, ki uporabijo različne načine računanja podobnosti med kromami. V sedmem poglavju testiramo predstavljene algoritme na arhivu slovenske ljudske glasbe za tri različne scenarije načina poizvedb. Ugotovimo, da predstavljene metode pomembno izboljšajo rezultate osnovnega algoritma za iskanje po arhivu ljudske glasbe. Magistrsko delo sklenemo s poglavjem o sklepnih ugotovitvah, v katerem predlagamo nekaj možnosti za nadaljnje delo.

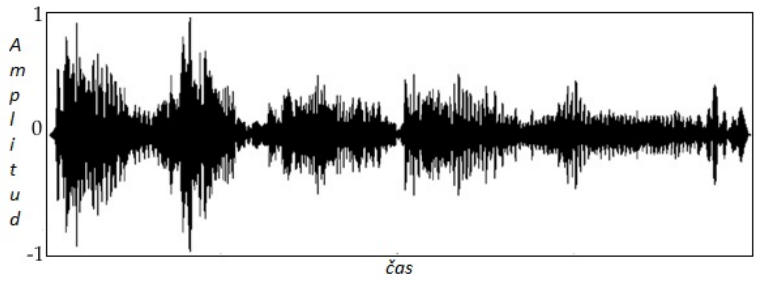
Poglavje 2

Vrste glasbenih zapisov

V tem poglavju predstavimo različne vrste glasbenih zapisov. Na podlagi poznavanja lastnosti posameznih glasbenih zapisov bomo v nadaljevanju lažje implementirali metode za poravnavo različnih vrst glasbenih zapisov.

2.1 Zvočni zapis glasbe

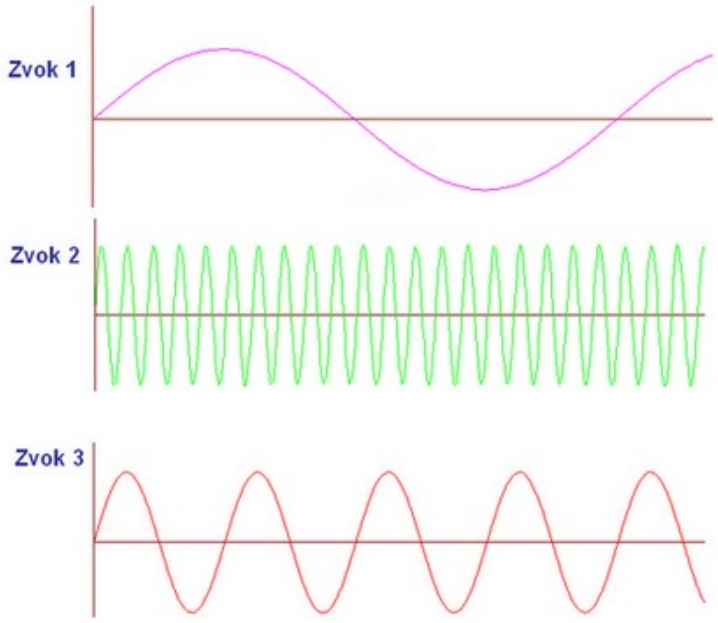
S stališča fizike je zvok mehansko valovanje, ki se širi v dani snovi. Zvok lahko grafično predstavimo kot funkcijo amplitude v odvisnosti od časa (slika 2.1).



Slika 2.1: Grafična predstavitev zvoka.

Zgornja predstavitev pa nam ne pove kaj dosti o lastnostih zvoka. Vemo, da zvok doživljamo kot različno visok, nizek, glasen, tih. Pri višini zvoka je pomemben koncept frekvence. Frekvenca nam pove, kolikokrat se neki

dogodek ponovi v enoti časa. Za tako imenovane čiste tone ali sinusne valove, ki so prikazani na sliki 2.2, lahko določimo frekvenco tona.



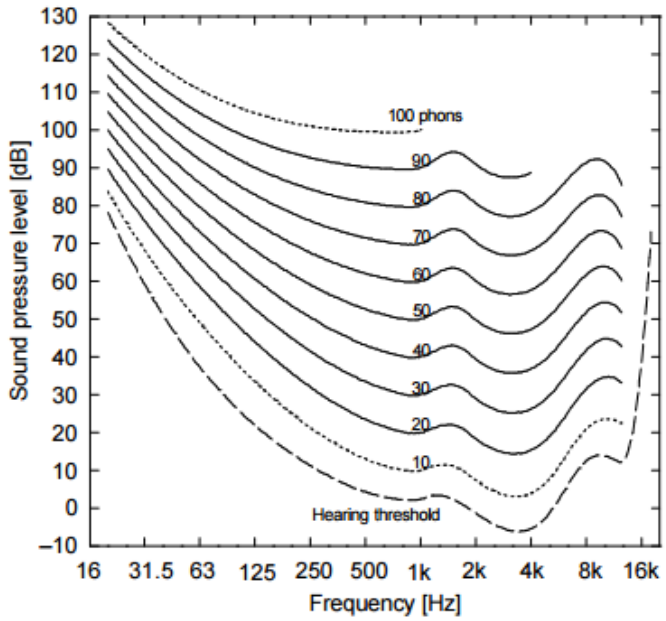
Slika 2.2: Sinusni zvoki z različno frekvenco.

Kadar z glasom zapojemo neki ton, ne nastane enostavna sinusna krivulja, ampak nastane več sinusnih krivulj - harmonikov. Harmonik z najnižjo frekvenco se imenuje osnovni harmonik in določa višino tona, ostali harmoniki pa dajejo tonu specifično barvo.

Pri percepciji zvoka je potrebno omeniti še princip intervalov, oziroma razdalj med toni. Še posebej pomemben je interval oktave, saj ljudje dva tona, ki sta oddaljena za oktavo, dojemamo kot sorodna. Ta razdalja pa ni premosorazmerna s frekvenco, ampak bazira na logaritmu z osnovo 2. Sorodni so na primer toni s frekvenco 400, 800, 1600 Hz.

Poleg višine tona je pomembna tudi njegova glasnost. Glasnost zvoka je močno korelirana z jakostjo zvoka, ki je fizikalna količina, določena kot gostota energijskega toka zvočnega valovanja. Potrebno je dodati, da je percepcija glasnosti pri ljudeh odvisna ne le od jakosti tona, ampak tudi od

njegove višine, kot je prikazano na sliki 2.3.

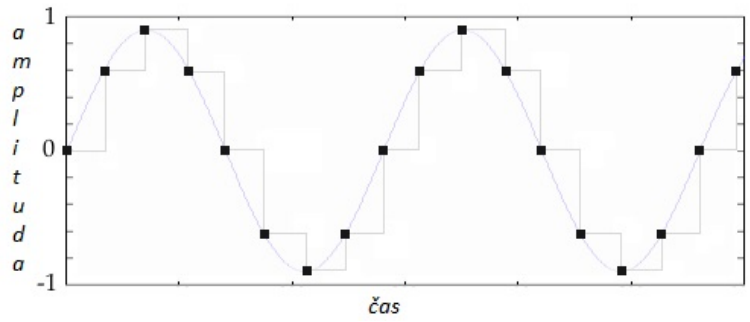


Slika 2.3: Prikaz percepcije glasnosti v odvisnosti od frekvence. Točke na isti črti zaznavamo kot enako glasne. Slika je iz vira [16].

Zvok je zvezen pojav, zato ga je za računalniško obdelavo potrebno najprej prevesti v diskretno obliko. To storimo s tako imenovanim vzorčenjem, kot je prikazano na sliki 2.4, tako da na enako oddaljenih časovnih točkah zabeležimo kvantizirano amplitudo zvoka [17]. Pri tem izgubimo del informacije o zvoku, vendar je ob ustrezni frekvenci vzorčenja (npr. 44.100 Hz) ter primerni kvantizaciji ta del minimalen s stališča človeške percepcije zvoka.

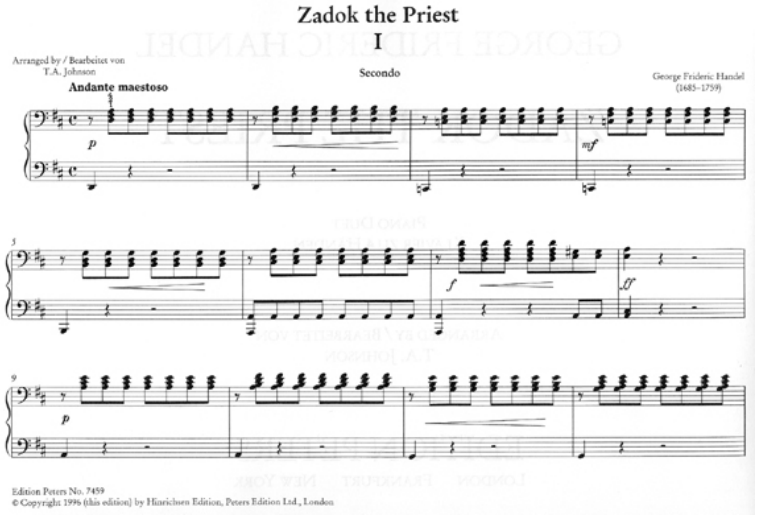
2.2 Notni zapis glasbe

Notni zapis predstavlja simbolni zapis glasbe. S simboli označimo tempo, glasnost, dinamiko in artikulacijo glasbe ter za posamezen ton njegov začetek, trajanje in višino. S pomočjo notnega zapisa se lahko pevci ali instrumentalisti naučijo nove skladbe in jo izvedejo. Napotki za izvajanje še vedno



Slika 2.4: Prikaz vzorčenja analognega posnetka. Digitalni posnetek vsebuje le s piko označene podatke o velikosti amplitude v določenem trenutku.

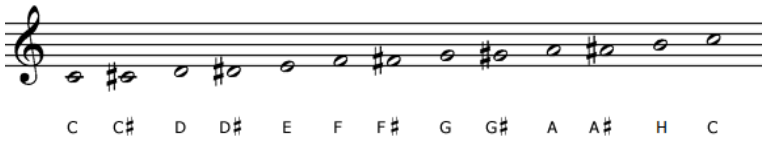
dopuščajo izvajalcu veliko svobode pri interpretaciji dela, kar vodi v raznolike izvedbe istega notnega zapisa. Primer notnega zapisa je prikazan na sliki 2.5.



Slika 2.5: Primer notnega zapisa.

Osnovna sestavina notnega zapisa so glasbene note. Glasbene note predstavljajo relativno dolžino in višino zvoka. Dve noti z osnovno frekvenco v razmerju potence števila 2 (na primer polovica, dva, štiri) zaznamo kot soro-

dni. Zato lahko note grupiramo v posamezne razrede. V glasbeni teoriji se uporablja naslednje poimenovanje za glasbene note: C, Cis (Es), D, Dis (Es), E, F, Fis(Ges), G, Gis(As), A, Ais(B), H. Različne notne višine so prikazane na sliki 2.6.

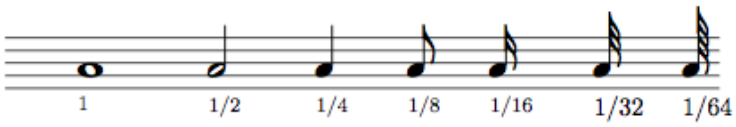


Slika 2.6: Različne notne višine.

Note se med seboj razlikujejo tudi po trajanju. Izhodišče nam je nota celinka. Ta nam relativno označuje najdaljši ton. Iz celinke so po delitvah izpeljane druge notne vrednosti, ki imajo ime po številki v ulomku. Glede na trajanje ločimo torej naslednje note:

- celinke (1),
- polovinke (1/2),
- četrtinke (1/4),
- osminke (1/8),
- šestnajstinke (1/16),
- dvaintridesetinke (1/32),
- štiriinšestdesetinke (1/64).

Izgled not, ki označujejejo različno trajanje tonov, je prikazan na sliki 2.7.



Slika 2.7: Različne notne dolžine.

2.3 Zapis glasbe MusicXML

MusicXML je zapis XML, zasnovan z namenom predstavitve notnih zapisov zahodne glasbe v formatu XML [18]. To je tekstovni zapis, ki vsebuje vse napotke za notno predstavitev. Kot vsi formati XML je tudi MusicXML namenjen enostavnemu razčlenjevanju in manipulaciji s strani avtomatiziranih orodij. S pomočjo formata MusicXML lahko zapišemo višino tona, trajanje, loke, dinamiko, okraske ... MusicXML je podprt v mnogih programih, namenjenih pisanju not, branju ročno zapisanih not in snemanju ter obdelavi zvočnih posnetkov. S pomočjo MusicXML lahko enostavno prenašamo glasbeni zapis med omenjenimi programi. Primer zapisa MusicXML je prikazan na sliki 2.8.

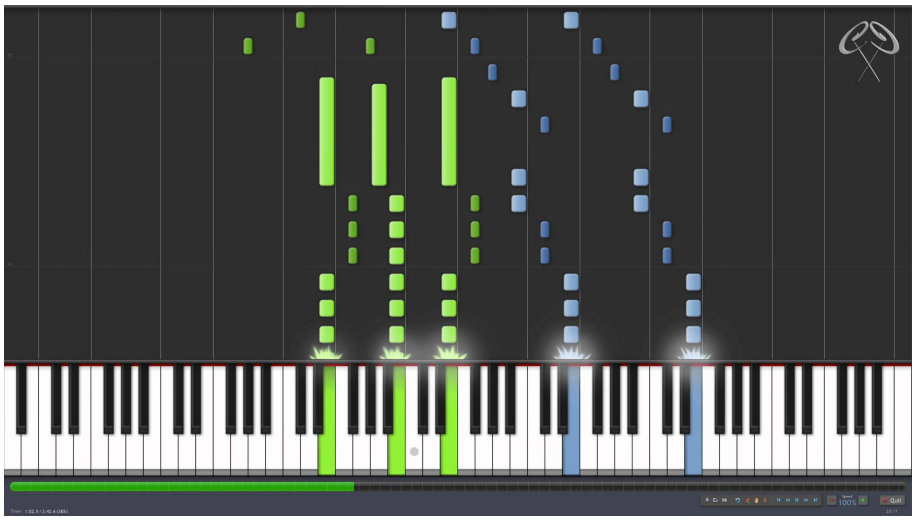
```
<note>
  <pitch>
    <step>C</step>
    <octave>4</octave>
  </pitch>
  <duration>4</duration>
  <type>whole</type>
</note>
```

Slika 2.8: Primer zapisa MusicXML za noto C4.

2.4 Zapis glasbe MIDI

Zapis MIDI je simbolični zapis glasbe, ki je bil sprva razvit kot standard za medsebojno komunikacijo in delovanje digitalnih glasbenih inštrumentov[19].

Podatke MIDI lahko tudi posnamemo in shranimo za poznejše urejanje ter predvajanje. Zapis MIDI vsebuje informacije o začetku in koncu določenega tona, o njegovi višini ter o njegovi moči oziroma glasnosti. Dodatno vsebuje še podatek o kanalu oziroma vrsti inštrumenta. Zapis MIDI nam lahko posreduje več informacij glede interpretacije kot sam notni zapis, kljub vsemu pa je zapis MIDI omejen, predvsem kar se tiče barve zvoka. Grafična predstavitev zapisa MIDI je prikazana na sliki 2.9.



Slika 2.9: Vizualna predstavitev zapisa MIDI kot rezultat igranja na električni klavir.

Ker je zapis MIDI zaporedje ukazov, nam omogoča spreminjanje ključa, tempa ter inštrumentov, kar pri posnetem zvočnem zapisu ni mogoče. Obstajajo programi, ki pretvorijo zapis MIDI v notni zapis glasbe, kar omogoča hitrejše zapisovanje not. Pomembno aplikacijo zapisa MIDI predstavljajo še programi za avtomatsko spremljavo živega izvajanja glasbe, ki na podlagi žive glasbe ustvarijo spremljavo v obliki zapisa MIDI in jo posredujejo napravi za generiranje zvoka.

Poglavje 3

Tonske in kromatične zvočne značilke

V prejšnjem poglavju smo predstavili osnovne tipe glasbenih zapisov. Da bi omenjene zapise lahko primerjali, je najprej potrebno uvesti neko vmesno predstavitev glasbe, na podlagi katere bi lahko primerjali osnovne glasbene zapise. Te vmesne značilke morajo čim bolj izražati pomembne lastnosti glasbe, medtem ko morajo izvzeti nepomembne dele oziroma dele, ki se spreminjajo glede na interpretacijo oziroma izvedbo dela.

Kot pomembna lastnost za naš namen se izkaže tonska predstavitev glasbe, v kateri predstavimo glasbo kot časovno zaporedje tonov zahodne lestvice. Na podlagi te predstavitve izpeljemo kromatično predstavitev glasbe, ki predstavlja časovno zaporedje dvanajstdimenzionalnih vektorjev, pri čemer vsaka dimenzija predstavlja ton dvanajsttonske kromatične lestvice. Postopek povzamemo po [20].

3.1 Tonske značilke

Tonsko značilko si lahko predstavljamo kot vektor, ki vsebuje vrednosti za vseh 88 tonov, kot jih vsebuje standarden klavir. Ekstrakcija tonskih značilk

poteka s pomočjo pasovnih filtrov, ki prepuščajo frekvence okoli določenega osnovnega tona, medtem ko utišajo ostali del signala, ki je izven pasa frekvenčnega spektra. Pri tem morajo imeti filtri ozek prepustni del, strm prehod in dobro zavračanje izven prepustnega dela. Prav tako red filtrov ne sme biti prevelik za učinkovito računanje. V naši aplikaciji bomo uporabili eliptične filtre. Filtre za nizke tone apliciramo na signal z nižjo frekvenco vzorčenja (npr. 882 Hz), medtem ko filtre za srednje in visoke tone apliciramo na signal z višjo frekvenco vzorčenja (npr. 4.410 Hz za srednje tone in 22.050 Hz za visoke tone). Zmanjšanje frekvence vzorčenja pri nizkih tonih prispeva tudi k pohitritvi računanja.

3.2 Značilke lokalne energije

Na podlagi algoritma za izračun tonskih značilk dobimo 88 različnih signalov za vsakega izmed osnovnih tonov. Če z x označimo signal i-tega osnovnega tona ter z w širino okna, potem lahko energijo pri času n izračunamo s pomočjo formule 3.1.

$$e_{n,i} = \sum_{k \in [n-\frac{w}{2} : n+\frac{w}{2}]} |x(k)|^2 \tag{3.1}$$

Kvadratno okno lahko nadomestimo tudi s kakšnim primernejšim oknom, kot je na primer Hannovo okno ali trikotno okno.

Da zmanjšamo število podatkov, lahko lokalne značilke računamo vsakih d vzorcev in na ta način za faktor d zmanjšamo količino podatkov.

3.3 Kromatične značilke

Na podlagi značilk lokalne energije za pripadajoče tonske značilke, lahko sedaj dobimo kromatične značilke. Postopek povzamemo po [20]. Kromatične

značilke so koristne zaradi dejstva, da različni inštrumenti proizvajajo zvok, ki vsebuje harmonike v različnih razmerjih, kar povzroči različne barve zvoka. Kromatične značilke nadgradijo tonske značilke na način, da seštejejo tonske spektre tonov, ki pripadajo istemu tonu kromatične lestvice. Kromatične značilke so zato robustne na barvo zvoka. Tako na primer kromagram za ton C dobimo na način, da seštejemo značilke lokalne energije za tone C0, C1, C2, ... , C8, kar storimo s pomočjo formule 3.2. S seštevanjem značilk lokalne energije tako dobimo dvanajstdimenzionalne vektorje oziroma krome, vsaka dimenzija vektorja pa pripada enemu izmed dvanajstih tonov kromatične lestvice.

$$v_{n,i} = \sum_{j \ mod \ 12=i} e_{n,j} \tag{3.2}$$

Kromatične značilke lahko dodatno izboljšamo z uvedbo normalizacije krom. Na ta način postanejo kromatične značilke robustne na spremembe v dinamiki izvajanja glasbe. Krome normaliziramo na način, da vektor v nadomestimo z vektorjem $v/||v||_1$, pri čemer velja $||v||_1 = \sum_{i=0}^{11} |v(i)|$.

Naslednja izboljšava je kvantizacija amplitud kromagramov. Pri kvantizaciji upoštevamo logaritemsko naravo zaznave glasnosti zvoka, kar pomeni, da so spodnji razredi ožji. Tako lahko za kvantizacijsko funkcijo vzamemo na primer funkcijo K, definirano v 3.3.

$$K(a) = \begin{cases} 0 & \text{za } 0 \leq a < 0,05 \\ 1 & \text{za } 0,05 \leq a < 0,1 \\ 2 & \text{za } 0,1 \leq a < 0,2 \\ 3 & \text{za } 0,2 \leq a < 0,4 \\ 4 & \text{za } 0,4 \leq a < 1 \end{cases} \tag{3.3}$$

Na ta način lahko vsako kromo $v_n = (v_{n,0}, v_{n,1}, \ldots, v_{n,11}) \in [0,1]^{12}$ prevedemo v obliko

$$k_n := (K(v_{n,0}), K(v_{n,1}), \ldots, K(v_{n,11})) \tag{3.4}$$

Poglavje 4

Algoritmi za poravnavo glasbenih zapisov

Za implementacijo iskanja po arhivu ljudske glasbe moramo izvesti primerjavo med vzorčnim zapisom in vsemi kandidati. Pri tem si lahko pomagamo s časovno poravnavo parov glasbenih zapisov. Na podlagi časovno poravnanih zaporedij lahko nato računamo podobnost med zaporedji. V tem poglavju predstavimo dva najpomembnejša algoritma za poravnavo glasbenih zapisov, to sta algoritem dinamičnega časovnega krivljenja ter algoritem skritega modela Markova.

4.1 Dinamično časovno krivljenje

Dinamično časovno krivljenje uporabi paradigmo dinamičnega programiranja z namenom izračuna optimalne poravnave med dvema časovnima zaporedjema. Poravnava ukrivi eno izmed zaporedij na način, da se čim bolj prilega drugemu ter je osnova za izračun podobnosti med dvema zaporedjema. Dinamično časovno krivljenje je bilo najprej uporabljeno pri razpoznavi govora, pri nalogi določitve, ali dva posnetka besed predstavljata isto besedo. Predstavitev algoritma povzamemo po [20].

Algoritem primerja dva niza podatkov, X in Y, pri čemer sta oba niza oblike $X = (x_1, x_2, \ldots, x_n)$. Poljuben element x_i ali y_i se nahaja v prostoru značilk F ($x_i, y_i \in F$). Da bi lahko primerjali dva različna elementa $x_i \in X$, $y_i \in Y$, moramo definirati lokalno funkcijo razdalje c (formula 4.1).

$$c : F \times F \to \mathbb{R}_{\geq 0}. \tag{4.1}$$

Tipično je c majhna za podobni značilki x in y ter velika za zelo različni značilki. Z izračunom vseh možnih kombinacij lokalnih razdalj med različnimi vektorji iz X in Y dobimo matriko razdalj $C \in \mathbb{R}^{N \times M}$, kjer velja $C(n, m) := c(x_n, y_m)$. Naslednji cilj je najti optimalno poravnavo med X in Y, to je najti pot vzdolž matrike C, ki minimizira vsoto lokalnih razdalj.

Da bi lahko našli optimalno poravnavo med dvema zaporedjema X in Y, je najprej potrebno natančno definirati lastnosti poravnave. Naj bo torej poravnava $p = (p_1, \ldots, p_l)$ ter $p_l = (n_l, m_l) \in [1 : N] \times [1 : M]$ za $l \in [1 : L]$.

Poravnavo p med dvema zaporedjema X in Y mora zadoščati naslednjim pogojem:

- robni pogoj: $p_1 = (1, 1)$ in $p_L = (N, M)$,
- pogoj monotonosti: $n_1 \leq n_2 \leq \ldots \leq n_L$ in $m_1 \leq m_2 \leq \ldots \leq m_L$,
- pogoj velikosti koraka: $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}$ za $l \in [1 : L-1]$.

Enostavno je opaziti, da drugi pogoj sledi iz tretjega pogoja o velikosti koraka, torej je naveden le zaradi jasnosti definicije. Ukrivljena pot $p = (p_1, \ldots, p_L)$ definira poravnavo med dvema zaporedjema $X = (x_1, x_2, \ldots, x_N)$ ter $Y = (y_1, y_2, \ldots, y_N)$ na način, da določi dvojice $(x_{n_l}, y_{m_l}) \in (X \times Y)$. Robni pogoj vsili zahtevo, da sta prva elementa zaporedij med seboj poravnana ter enako za zadnja elementa zaporedij. Pogoj monotonosti pomeni, da poravnava ohranja časovno razvrstitev elementov. Pogoj velikosti koraka onemogoči ponavljanje identičnih elementov v poravnavi (korak ne more biti $(0, 0)$) ter zagotovi, da ni noben element posameznega zaporedja izpuščen v

poravnavi (premik v posameznem zaporedju ne sme biti večji od ena).

Sedaj lahko s formulo 4.2 definiramo skupno razdaljo $c_p(X, Y)$ glede na ukrivljeno pot p med dvema zaporedjema X in Y.

$$c_p(X, Y) := \sum_{l=1}^{L} c(x_{n_l}, y_{m_l}). \tag{4.2}$$

Na ta način lahko določimo optimalno pot p^* med dvema zaporedjema X in Y kot pot, ki ima minimalno skupno razdaljo izmed vseh možnih poti. Skupna razdalja c_{p^*} optimalne poti predstavlja optimalno razdaljo $DTW(X, Y)$ algoritma dinamičnega časovnega krivljenja za zaporedji X in Y (formula 4.3).

$$DTW(X, Y) := c_{p^*}(X, Y) = min\{c_p(X, Y)|\ p \text{ je ukrivljena pot med } X \text{ in } Y\} \tag{4.3}$$

Pomembno je poudariti, da razdalja DTW ne zagotavlja trikotniške neenakosti, četudi je razdalja c metrična razdalja.

Za določitev optimalne poti p^* bi lahko preizkusili vse možne ukrivljene poti med X in Y, a bi to vodilo do algoritma s časovno zahtevnostjo, ki bi bila eksponentna glede na dolžini X in Y. Zato se poslužimo dinamičnega programiranja, s pomočjo katerega dosežemo algoritem s časovno zahtevnostjo velikosti $O(NM)$.

Najprej za zaporedji X in Y definiramo podzaporedja, $X(1 : n) := (x_1, \ldots, x_n)$ ter $Y(1 : n) := (y_1, \ldots, y_n)$. Nato definiramo matriko D velikosti $N \times M$, ki jo poimenujemo akumulacijska matrika razdalj (formula 4.4).

$$D(n, m) := DTW(X(1 : n), Y(1 : m)). \tag{4.4}$$

Očitno velja $D(N, M) = DTW(X, Y)$. Enostavno je videti, da velja tudi $D(n, 1) = \sum_{k=1}^{n} c(x_k, y_1)$ za $n \in [1 : N]$ ter $D(1, m) = \sum_{k=1}^{m} c(x_1, y_k)$ za

$m \in [1 : M]$. Pokažemo pa lahko tudi, da velja:

$$D(n,m) = min\{D(n-1,m-1), D(n-1,m), D(n,m-1)\} + c(x_n, y_m) \quad (4.5)$$

za $1 < n \leq N$ in $1 < m \leq M$. Naj bosta torej $n > 1$ in $m > 1$ ter naj bo $q = (q_1, \ldots, q_{l-1}, q_l)$ optimalno ukrivljena pot za podzaporedji $X(1 : n)$ in $Y(1 : m)$. Iz robnega pogoja sledi $q_l = (n, m)$. Naj bo $(a, b) := q_{L-1}$. Iz pogoja o velikosti koraka sledi $(a, b) \in \{(n-1, m-1), (n-1, m), (n, m-1)\}$. Nadalje velja, $(q_1, \ldots, q_{l-1})$ mora biti optimalna ukrivljena pot za $X(1 : a)$ in $Y(1 : b)$, sicer q ne bi bila optimalna za osnovno podzaporedje $X(1 : n)$ in $Y(1 : m)$. Ker $D(n, m) = c_{q_1,\ldots,q_{l-1}}(X(1 : a), Y(1 : b)) + c(x_n, y_m)$ ter je q optimalna, smo dokazali enačbo 4.5.

Na podlagi enačbe 4.5 lahko rekurzivno izračunamo vrednosti akumulacijske matrike D. Inicializacijo lahko poenostavimo s povečanjem matrike D za dodatno vrstico in kolono ter določitvijo $D(n, 0) := \infty$ za $n \in [1 : N]$, $D(0, m) := \infty$ za $m \in [1 : M]$ in $m \in [1 : M]$. Opazimo tudi, da lahko D računamo po zaporednih stolpcih, kjer za izračun m-tega stolpca rabimo le vrednosti stolpca na položaju $m - 1$. Pomeni, da lahko ob predpostavki, da nas zanima le vrednost $DTW(X, Y)$, hranimo le vrednosti reda $O(n)$. Seveda lahko isti postopek izvedemo tudi po vrsticah. Kljub temu je časovna zahtevnost še vedno $O(NM)$ v obeh primerih. Če hočemo izračunati optimalno ukrivljeno pot p^*, potrebujemo celotno matriko D.

4.1.1 Dinamično časovno krivljenje za podzaporedja

V mnogo primerih imamo za cilj najti podzaporedje v nekem zaporedju, ki najbolj ustreza nekemu osnovnemu zaporedju. Lahko imamo na primer dan daljši posnetek, v katerem hočemo najti kratko melodijo. Iskanje optimalnega podzaporedja lahko rešimo z različico dinamičnega časovnega krivljenja, kot je predstavljeno v [20].

Naj bosta dani zaporedji $X := (x_1, x_2, \ldots, x_N)$ ter $Y := (y_1, y_2, \ldots, y_M)$ ter naj bo M mnogo večji od N. Cilj je najti podzaporedje $Y(a^* : b^*) :=$

$(y_{a^*}, y_{a^*+1}, \ldots, y_{b^*})$ z $1 \leq a^* \leq b^* \leq M$, ki minimizira razdaljo DTW do X preko vseh možnih podzaporedij Y. Velja formula 4.6.

$$(a^*, b^*) := argmin_{(a,b):1 \leq a \leq b \leq M}(DTW(X, Y(a:b))) \tag{4.6}$$

Števili a^*, b^*, kot tudi optimalno poravnavo med X in podzaporedjem $Y(a^*, b^*)$, lahko izračunamo s pomočjo modifikacije osnovnega algoritma dinamičnega časovnega krivljenja. To storimo na način, da ne kaznujemo preskokov poravnave med X in Y, ki se pojavijo na začetku in koncu zaporedja Y. S formulo lahko to spremembo opišemo na način, da velja $D(1, m) := c(x_1, y_m)$, namesto $D(1, m) := \sum_{k=1}^{m} c(x_1, y_k)$ kot v osnovnem algoritmu. Kot v osnovnem algoritmu velja $D(n, 1) := \sum_{k=1}^{n} c(x_k, y_1)$ za $n \in [1 : n]$ in ostale vrednosti matrike D za $n \in [2 : N]$ in $m \in [2 : M]$ izračunamo po rekurzivni formuli 4.5. Podobno kot pri osnovnem algoritmu dinamičnega časovnega krivljenja definiramo razširjeno akumulacijsko matriko D. Pri tem za razliko od osnovnega algoritma velja $D(0, m) := 0$, namesto $D(0, m) = \infty$ za $m \in [0 : M]$. Število $DTW(X, Y)$, ki predstavlja minimalno razdaljo med zaporedjema X in Y, določimo iz matrike D s formulo 4.7.

$$DTW(X, Y) := min_{b \in [1:M]} D(N, b) \tag{4.7}$$

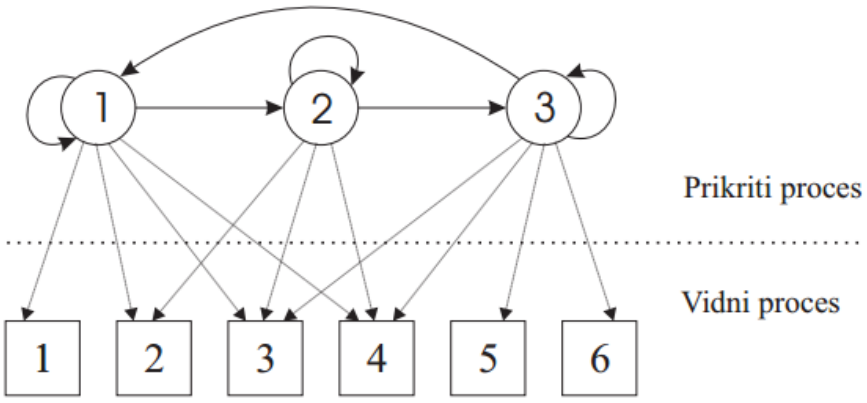
Očitno velja, da je časovna zahtevnost omenjenega algoritma velikosti $O(NM)$.

4.2 Skriti model Markova

4.2.1 Opis skritega modela Markova

Markov proces je statističen model, ki zadošča markovski lastnosti. Markovska lastnost določa, da so prihodnje vrednosti sistema odvisne le od trenutnega stanja sistema. Skriti model Markova (HMM) predstavlja Markov proces, pri katerem je zaporedje stanj sistema skrito, viden izhod sistema pa je verjetnostna funkcija stanja. Primer modela je predstavljen na sliki 4.1.

Opis sistema povzamemo po delih [21] in [22].



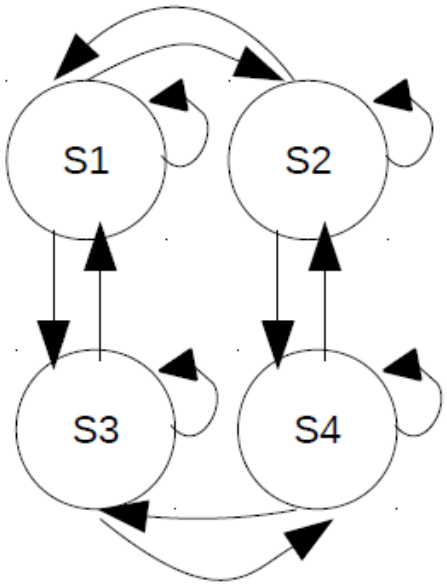
Slika 4.1: Primer modela HMM s tremi stanji in šestimi izhodnimi simboli.

Modele HMM lahko razdelimo v več različnih skupin glede na matriko prehodov, to je dovoljenih povezav med stanji. Naj bo $S = \{S_1, S_2, ..., S_N\}$ množica vseh različnih stanj sistema ter naj bo $A = \{a_{ij}\}$ matrika verjetnosti prehodov med posameznimi stanji, na primer med S_i in S_j. Prvi tip modela HMM je ergodičen (slika 4.2). V tem primeru obstaja število N, tako da je možno poljubno stanje sistema doseči iz poljubnega drugega stanja v natanko N korakih. Poseben primer ergodičnega modela HMM je popolnoma povezan model, kjer je možno poljubno stanje doseči iz poljubnega drugega stanja v enem koraku, oziroma $a_{ij} > 0$ za poljubna indeksa i, j.

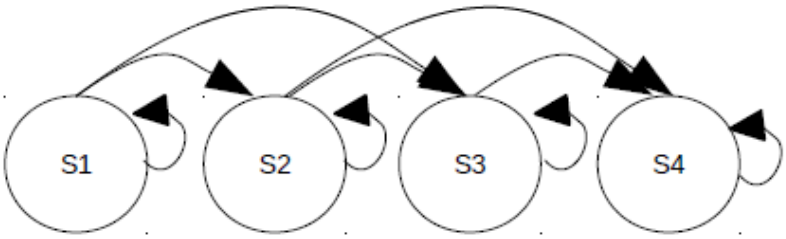
Naslednja skupina so Bakisovi modeli, pri katerih so možni prehodi le od leve proti desni. To pomeni, da niso možni prehodi iz trenutnega stanja k stanju z nižjim indeksom, oziroma $a_{ij} = 0, j < i$ (tabela 4.1).

Obstaja več variacij modelov od leve proti desni, ena od njih se imenuje model skokov (slika 4.3), ki omejuje, kako daleč naprej se lahko premaknemo iz trenutnega stanja in določa $a_{ij} = 0$ za $j > i + \delta$, kjer je δ maksimalna velikost skoka (tabela 4.2).

Sedaj, ko smo opisali strukturo skritega dela modela HMM, si oglejmo še



Slika 4.2: Primer ergodične markovske verige.



Slika 4.3: Primer markovske verige s skokom.

verjetnostne funkcije izhodov. Pri diskretnih modelih HMM imamo omejeno število možnih izhodov $V = \{V_1, V_2, ..., V_M\}$. Za vsako stanje S_i imamo podano verjetnostno funkcijo $b_i(k)$, ki nam pove verjetnost izhoda V_k, če se nahajamo v stanju S_i. Za uporabo v poravnavi glasbenih zapisov je boljši zvezni model HMM, pri katerem je množica izhodov oblike $V = \mathbb{R}^M$. Za vsako stanje S_i imamo sedaj podano zvezno porazdelitev $\phi_i : \mathbb{R}^m \to \mathbb{R}$. Za število stanj $N = 3$ in dimenzijo prostora izhodov $M = 1$ je primer zveznega

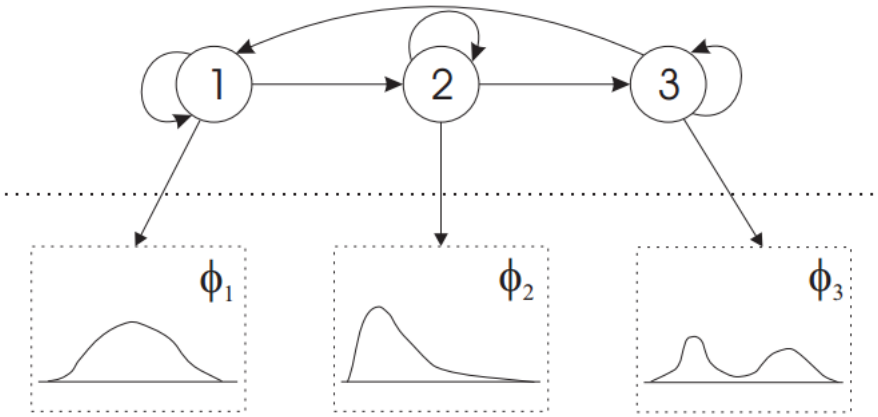
	S_1	S_2	S_3	S_4
S_1	a_{11}	a_{12}	a_{13}	a_{14}
S_2	0	a_{22}	a_{23}	a_{24}
S_3	0	0	a_{33}	a_{34}
S_4	0	0	0	a_{44}

Tabela 4.1: Tabela prehodov za Bakisov model prehodov med stanji.

	S_1	S_2	S_3	S_4
S_1	a_{11}	a_{12}	a_{13}	0
S_2	0	a_{22}	a_{23}	a_{24}
S_3	0	0	a_{33}	a_{34}
S_4	0	0	0	a_{44}

Tabela 4.2: Tabela prehodov za model skokov, kjej je δ enak 2.

modela HMM predstavljen na sliki 4.4.



Slika 4.4: Primer zveznega modela HMM.

4.2.2 Osnovni problemi modelov HMM

Sedaj, ko smo predstavili modele HMM, nadaljujemo s tremi osnovnimi problemi, ki nas zanimajo pri modelih HMM:

- Problem ocenjevanja. Podan imamo model HMM λ in zaporedje opazovanj $O = o_1, o_2, ..., o_T$. Zanima nas, kakšna je verjetnost opazovanj glede na naš model, to je $p(O|\lambda)$.

- Problem dekodiranja. Podan imamo model HMM λ in zaporedje opazovanj $O = o_1, o_2, ..., o_T$. Zanima nas najbolj verjetno zaporedje stanj, skozi katera je model prehajal, da je tvoril to zaporedje opazovanj, oziroma $S = \underset{S'}{\operatorname{argmax}} P(S'|O, \lambda)$.

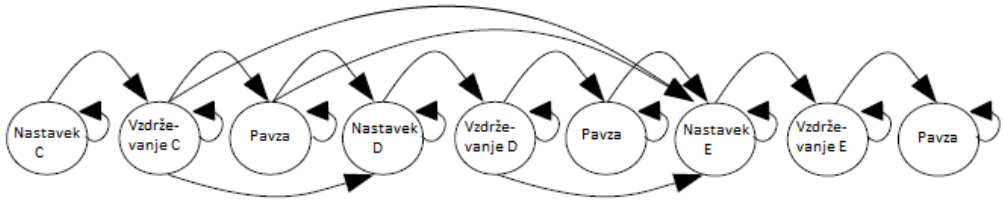
- Problem učenja. Podano imamo zaporedje opazovanj $O = o_1, o_2, ..., o_T$. Zanima nas, kako moramo prilagoditi parametre modela HMM λ, da bomo maksimizirali pogojno verjetnost $P(O|\lambda)$.

Pri reševanju omenjenih problemov si pomagamo z različnimi algoritmi. Za reševanje problema ocenjevanja uporabimo algoritem naprej ali algoritem nazaj, za problem dekodiranja uporabimo Viterbijev algoritem, za problem učenja pa uporabimo Baum-Welchev algoritem. Opis algoritmov najdemo na primer v [23].

4.2.3 Uporaba modelov HMM za poravnavo glasbenih zapisov.

Podlago za model HMM predstavlja simbolni zapis glasbe, ki je lahko npr. v MIDI obliki. Za vsako noto zapisa ustvarimo tri stanja, in sicer nastavek, vzdrževanje in pavza. Nato določimo verjetnosti prehodov med vsemi ustvarjenimi stanji. V najbolj preprostem primeru določimo možnost prehoda le v trenutno in naslednje stanje ter omogočimo prehod iz posameznega stanja vzdrževanja v naslednje stanje nastavka, to je, omogočimo izpustitev pavze. Za modeliranje verjetnosti prehodov uporabimo enakomerno porazdelitev.

Opisana shema predvideva, da se simbolni zapis glasbe izvede brez napak. Z namenom večje robustnosti algoritma zato dodamo še možnost dodatnih prehodov med stanji. Omogočimo prehod iz posameznega stanja vzdrževanja ali pavze na drugi sledeči nastavek, za primer, da izvajalec izpusti naslednjo noto. Nov model prehodov med posameznimi stanji je predstavljen na sliki 4.5.



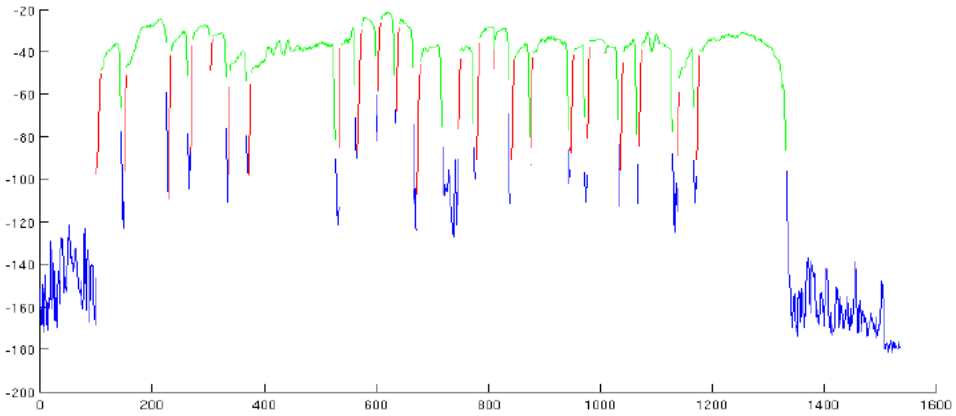
Slika 4.5: Primer modela prehodov med posameznimi stanji.

Sedaj, ko smo določili skriti model prehoda med posameznimi stanji, moramo določiti še verjetnost posameznih stanj v odvisnosti od izvajane glasbe, to je določiti moramo verjetnostne porazdelitve izhodnih funkcij. Za to je potrebno najprej na podlagi izvajane glasbe konstruirati posamezne zvočne značilke. Vsako od teh zvočnih značilk lahko nato povežemo z verjetnostjo nahajanja v posameznem stanju. Da bi to lahko storili, moramo najprej skozi proces učenja, v katerem analiziramo porazdelitve posameznih zvočnih značilk na vnaprej označenih zvočnih posnetkih. Primer označenega posnetka je prikazan na sliki 4.6.

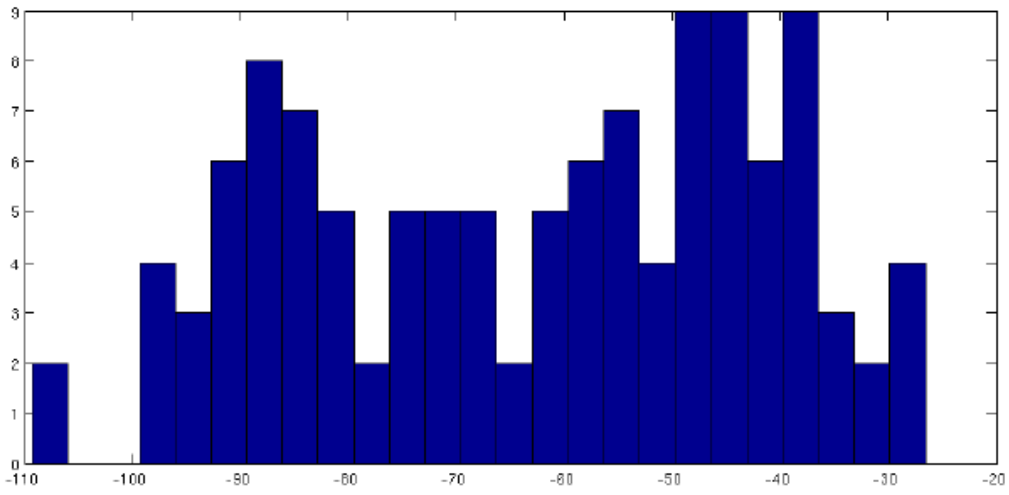
Na podlagi označenih posnetkov dobimo porazdelitve vrednosti zvočnih značilk za posamezna stanja, kot je prikazano na sliki 4.7 za stanje nastavka.

Na podlagi porazdelitev vrednosti zvočnih značilk za posamezna stanja dobimo Gaussove funkcije na način, da izračunamo srednjo vrednost in varianco. Za naš primer so Gaussove funkcije prikazane na sliki 4.8.

Dobljene funkcije uporabimo kot porazdelitvene funkcije izhodov za posamezna stanja pri HMM modelu. Več zvočnih značilk lahko za posamezno stanje kombiniramo na način, da za porazdelitveno funkcijo izhoda določimo

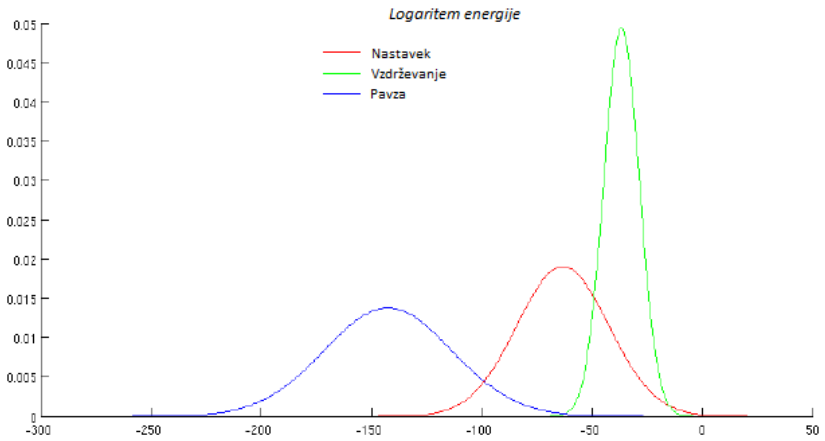


Slika 4.6: Primer značilk logaritma energije. Pavze so označene z modro, nastavki z rdečo, vzdrževanja tonov pa z zeleno barvo. Slika je iz vira [22].



Slika 4.7: Število pojavitev (navpična os) posamezne vrednosti značilke (vodoravna os) na vseh vzorcih, ki pripadajo stanju nastavka.

produkt posameznih funkcij zvočnih značilk.



Slika 4.8: Gaussove porazdelitve za tri različna stanja.

Poglavje 5

Analiza zvočnih zapisov ljudske glasbe

V magistrskem delu kot vir zvočnih zapisov uporabimo posnetke pete ljudske glasbe. Posnetki so izvajani s strani amaterskih pevcev, zato kvaliteta petja ni najboljša. Prisotne so napake v intonaciji, kar pomeni, da ton bodisi ni zapet točno, bodisi ni zapet pravi ton. Z namenom ustrezne predobdelave posnetkov, ki bi omogočila boljše delovanje našega algoritma za iskanje po zbirki posnetkov ljudske glasbe, je najprej potrebno opraviti analizo najpogostejših odstopanj v posnetkih. S to analizo želimo nasloviti cilj magistrske naloge, ki je prilagoditev metod za poravnavo zvočnih in notnih zapisov na način, da izboljšamo njihovo delovanje na arhivu ljudske glasbe. Ta analiza nam pomaga razumeti težave osnovnih algoritmov za poravnavo glasbenih zapisov pri uporabi na arhivu ljudske glasbe in nam bo v nadaljevanju v pomoč pri načrtovanju ustreznih metod za izboljšanje osnovnih metod za poravnavo glasbenih zapisov.

Pri analizi smo primerjali kromagrame, pridobljene na osnovi zapisa MIDI s kromagrami, pridobljenimi z ljudskim petjem. Prisotne so bile naslednje vrste razlik oziroma napak:

- Prva razlika, ki jo opazimo, je prisotnost alikvotov oziroma harmonikov v kromagramu. Najbolj opazen je drugi harmonik, ki se v kromagramu

nahaja kvinto nad osnovnim tonom, to je 7 poltonov nad osnovnim tonom. Primer je prikazan na sliki 5.1a.

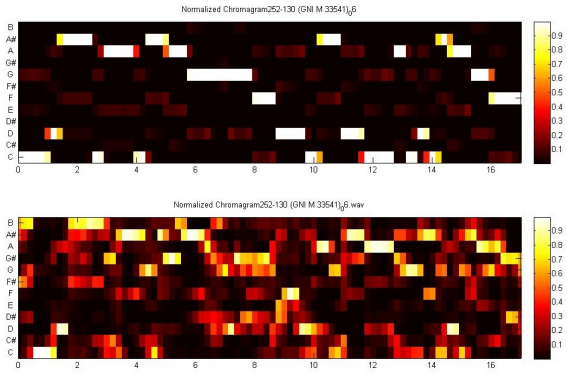
- Druga razlika je že posledica netočnosti petja, izraža pa se bodisi kot v celoti zgrešen ton, ki je v kromagramu za eno mesto nižje ali višje od pravega tona, bodisi kot deloma zgrešen ton, kar pomeni, da se je intonacija tona spreminjala. Primer je prikazan na sliki 5.1a.

- Pri nekaterih parih zapisa MIDI in zvočnega posnetka ljudske pesmi, kjer gre za dvoglasje ali večglasje, se pojavi razlika v moči oziroma glasnosti posameznega glasu pri zvočnem zapisu in zapisu MIDI. Tako sta glasova pri zapisu MIDI navadno izenačena po jakosti, pri zvočnem posnetku pa en glas pogosto močno prevladuje. Primer je prikazan na sliki 5.1b.

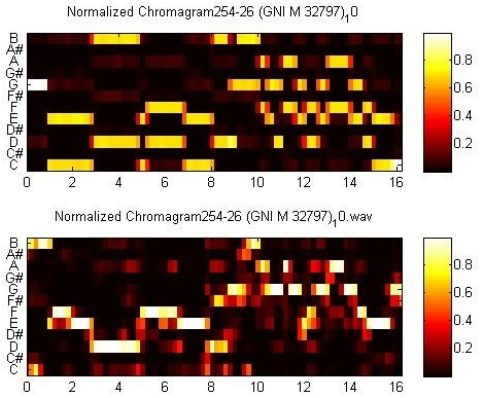
- Pri nekaterih parih zapisa MIDI in zvočnega posnetka ljudske pesmi, se v zvočnem posnetku pojavljajo kratki premori ali pavze, pri zapisu MIDI pa le teh ni. Primer je prikazan na sliki 5.1c.

- Ponekod so pri zvočnem posnetku prisotni deli, v katerih pevec bolj govori, kot poje besedilo, kar ima za posledico zelo razmazano kromo zvočnega posnetka, kroma zapisa MIDI pa prikazuje enoglasno petje. Primer je prikazan na sliki 5.1d.

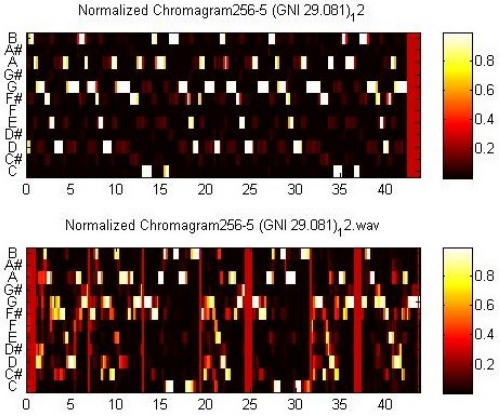
- Nekateri zvočni posnetki vsebujejo tudi primere tako imenovanega glissanda, ko glas drsi med dvema tonoma, zapisi MIDI pa vsebujejo le strogo ločene tone. Primer je prikazan na sliki 5.1e.



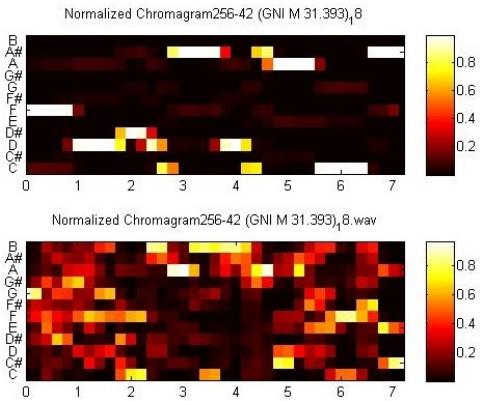
(a) Na začetku druge sekunde je lepo vidno odstopanje zvočne krome za eno mesto navzgor. Večkrat se pojavi šibek drugi harmonik.



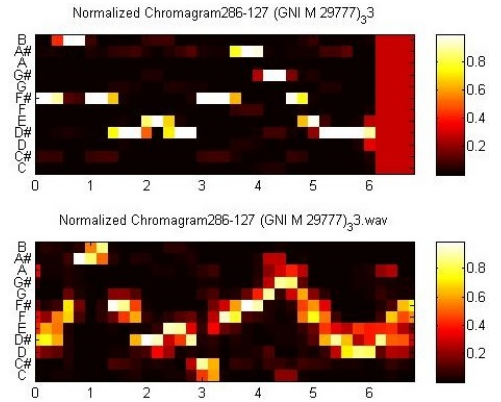
(b) Vidimo, da pri zvočni kromi en glas prevladuje, medtem ko sta pri kromi zapisa MIDI glasova izenačena.



(c) Kroma zvočnega posnetka vsebuje rdeče navpične črte, ki predstavljajo izenačenost frekvenčnega spektra, ta pa je posledica premorov pri izvajanju ljudske pesmi.



(d) Kroma zvočnega posnetka je na nekaterih delih precej razmazana, kar je posledica na pol govorjenega izvajanja skladbe.



(e) Primer prikazuje glissando na zvočni kromi okoli četrte sekunde.

Slika 5.1: Primerjava MIDI (zgoraj) in zvočnih (spodaj) kromagramov.

Poglavje 6

Algoritmi za iskanje po arhivu ljudske glasbe

V poglavju 4 smo že predstavili različne metode za poravnavo glasbenih zapisov ter v poglavju 5 analizirali značilnosti ljudskih glasbenih zapisov. Na podlagi tega lahko sprejmemo odločitev o optimalnem algoritmu, ki ga bomo v nadaljevanju uporabili za poravnavo ljudskih glasbenih zapisov. Odločitev utemeljimo v razdelku 6.1. V nadaljevanju nato v razdelku 6.2 podrobno opišemo implementacijo izbranega algoritma. V razdelku 6.3 naslovimo enega izmed osrednjih ciljev magistrske naloge, to je nadgradnja osnovnih algoritmov za iskanje po arhivih glasbe z namenov doseganja boljših rezultatov na arhivu ljudske glasbe.

6.1 Izbor metode za poravnavo glasbenih zapisov

V poglavju 1 smo predstavili različne algoritme, ki se uporabljajo pri nalogi iskanja v arhivu glasbe na podlagi petja. Pri tem smo navedli, da sta glede natančnosti iskanja najboljše rezultate dosegla algoritma DTW in HMM. V poglavju 4 smo predstavili omenjena algoritma. Pri algoritmu HMM smo spoznali, da je potrebno najprej natančno modelirati različne glasbene do-

godke. V poglavju 5 smo ugotovili, da posnetki ljudske glasbe pogosto vsebujejo različne vrste napak. Že v uvodu smo navedli, da so v članku [13] na podlagi testov ugotovili, da metoda HMM doseže večjo natančnost od metode DTW na zelo dobro zapetih poizvedbah, obraten rezultat pa je bil ugotovljen na množici slabše zapetih poizvedb. Glede na prej povedano za metodo poravnave izberemo metodo DTW.

6.2 Opis osnovnega algoritma za iskanje po arhivu ljudske glasbe

Vhod našega algoritma predstavljajo zvočni zapisi in zapisi MIDI slovenske ljudske glasbe. Najprej opišimo postopek pridobitve kromagrama določene slovenske ljudske pesmi iz ustreznega zvočnega posnetka.

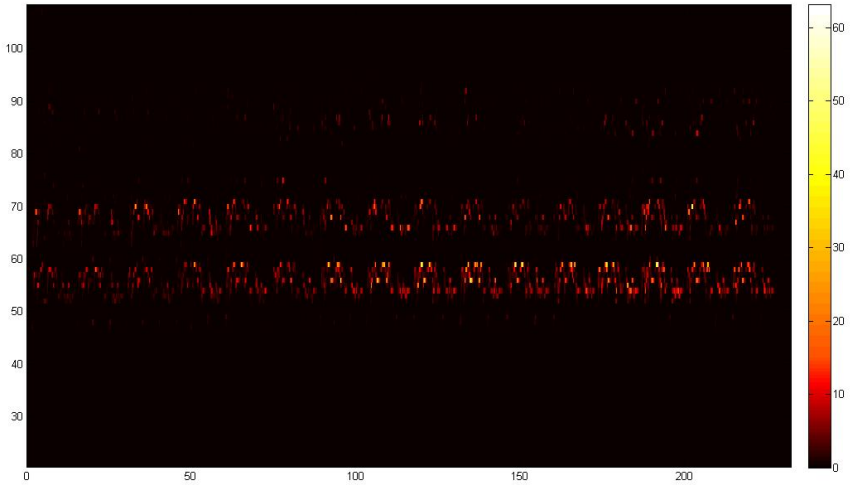
Na začetku preberemo zvočni posnetek ter po potrebi pretvorimo večkanalni signal v enokanalni signal. V naslednjem koraku izračunamo intonacijo zvočnega posnetka. To storimo na način, da preizkusimo različne intonacije za ton A4. Začnemo z osnovno intonacijo, ki znaša $440Hz$, nato pa se pomikamo za šestino poltona navzdol, dokler ne dosežemo naslednjega poltona, to je $G\#4$, ki ima frekvenco $415,3Hz$. Preizkusiti moramo torej šest intonacij $t_j = 440Hz - \frac{j}{6}(440 - 415,3)Hz$ za $j \in \{0, 1, \ldots, 5\}$.

Za vsako intonacijo izračunamo značilke lokalne energije, predstavljene v poglavju 3.2. Značilke lokalne energije izračunamo za 88 tonov standardnega klavirja, to je za tone A0 do C8 ter za celotno trajanje posnetka. Na ta način dobimo vrednosti $e_{t_j,n,i}$, kjer $n \in \{0, 1, \ldots, N\}$ predstavlja časovno komponento, $i \in \{0, 1, \ldots, 87\}$ pa tonsko komponento. Nato izračunamo vsoto značilk lokalne energije tako po časovni komponenti kot po tonski komponenti. Za zmagovalno proglasimo tisto intonacijo, ki maksimizira vsoto lokalne energije (formula 6.1). Na ta način dosežemo, da je kar največ frekvenčnega spektra signala v bližini tonov kromatične lestvice. Če bi se posamičen zapet ton nahajal med dvema tonoma kromatične lestvice, bi to namreč imelo za

posledico netočno kromo, saj bi algoritem za računanje krom ton pripisal obema najbližjima kromatičnima tonoma.

$$t_{max} = argmax_{t=t_0,t_1,\ldots,t_5} \sum_{n=0}^{N} \sum_{i=0}^{87} e_{t,n,i} \tag{6.1}$$

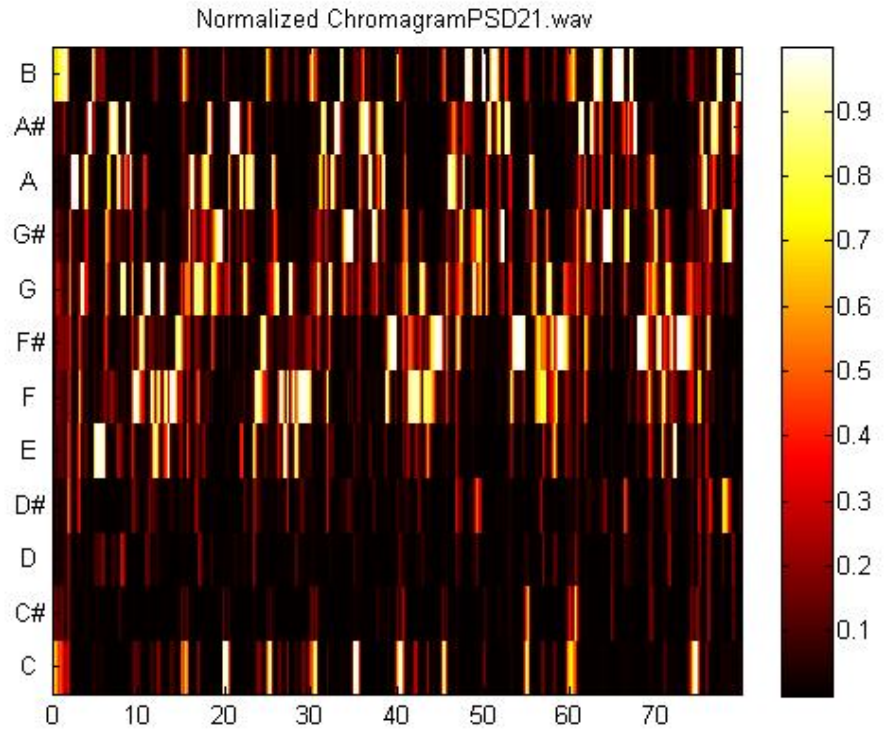
Na podlagi ugotovljene intonacije lahko za posamičen zvočni zapis določimo ustrezne značilke lokalne energije $e_{t_{max},n,i}$ (slika 6.1). Značilke lokalne energije določimo za tone v izbranem frekvenčnem območju, v našem algoritmu je to območje od 28 Hz do 4200 Hz, ki vsebuje tone A0 do C8.



Slika 6.1: Primer zaporedja značilk lokalne energije ljudske pesmi.

V zadnjem koraku na podlagi časovnega zaporedja značilk lokalne energije tvorimo časovno zaporedje krom (slika 6.2). To storimo na način, da seštejemo posamezne komponente značilk lokalne energije, ki pripadajo natanko za oktavo oddaljenim tonom. Na ta način dobimo zaporedje 12 dimenzionalnih krom. Na koncu še vsako kromo normaliziramo glede na L^1 normo in izvedemo kvantizacijo amplitud kromagramov, kot je predstavljeno v 3.3. Dobimo kromagram $k_{n,i}$, kjer $n \in \{0, 1, \ldots, N\}$ predstavlja časovno komponento, $i \in \{0, 1, \ldots, 11\}$ pa tonsko komponento. Pretvorba zapisa MIDI v

kromagram poteka na podoben način.



Slika 6.2: Primer kromagrama ljudske pesmi.

V naslednji fazi algoritma iščemo za posamezen kromagram zapisa MIDI, ki pripada neki ljudski pesmi, ustrezen kromagram zvočnega zapisa. Za primerjavo kromagramov uporabimo algoritem dinamičnega časovnega krivljenja. Še pred tem pa moramo zagotoviti ustrezen skupen ključ za posamezen par zvočnega kromagrama in kromagrama MIDI. To storimo na način, da za posamezen kromagram seštejemo krome vzdolž časovne komponente ter na ta način dobimo histogram, ki predstavlja tonsko sestavo ustreznega zvočnega zapisa ali zapisa MIDI (formula 6.2). Ta histogram nato še normaliziramo glede na L^1 normo.

$$h(i) = \sum_{n=0}^{N} k_{n,i}, \quad \text{i} = 0, 1, \ldots, 11. \tag{6.2}$$

V naslednjem koraku želimo za histogram zapisa MIDI določiti tako rotacijo histograma zvočnega zapisa, da bo razdalja med histogramoma minimalna. To storimo tako, da se sprehodimo po vseh možnih rotacijah zvočnega histograma ter pri vsaki iteraciji izračunamo razdaljo med histogramoma (formula 6.3) . Za računanje razdalje med histogramoma uporabimo evklidsko razdaljo. Izberemo tisto rotacijo, kjer je razdalja med histogramoma minimalna.

$$i = argmin_{i=0,\ldots,11} \sqrt{\sum_{j=0}^{11} (h_1(j) - h_2((j+i)\ mod\ 12))^2} \tag{6.3}$$

Končno lahko izračunamo razdaljo med dvema kromagramoma. Najprej zarotiramo drugi kromagram za ustrezno rotacijo, ki ustreza minimalni razdalji med pripadajočima histogramoma. Sedaj izvedemo algoritem dinamičnega časovnega krivljenja med prvim ter ustrezno zarotiranim drugim kromagramom. Algoritem dinamičnega časovnega krivljenja je zaradi učinkovitejšega delovanja implementiran v jeziku C. Algoritem vrne minimalno razdaljo med ustrezno ukrivljenima kromagramoma.

6.3 Opis izboljšav osnovnega algoritma za iskanje po arhivu ljudske glasbe

Namen izboljšav je izboljšati natančnost iskanja po arhivu ljudske glasbe. Pri načrtovanju izboljšav osnovnega algoritma za iskanje po arhivu ljudske glasbe se naslonimo na poglavje 5, v katerem opišemo nekatere značilne razlike med zvočnimi posnetki in zapisi MIDI. Prav tako pri načrtovanju izboljšav upoštevamo potek osnovnega algoritma za iskanje po arhivu ljudske glasbe ter njegove značilnosti.

6.3.1 Algoritem za razlikovanje med enoglasnim in večglasnim petjem

V našem arhivu ljudske glasbe so zbrani zapisi tako enoglasnega kot večglasnega petja. Pri iskanju ustreznih pripadajočih zapisov za neki določen zapis, bi bilo torej koristno vedeti, ali določen zapis predstavlja enoglasno ali večglasno petje. Na ta način lahko zožimo nabor možnih pripadajočih zapisov za posamezen zapis. Tako lahko za zapis MIDI za katerega algoritem smatra, da pripada enoglasnemu petju, iščemo ustrezne zadetke le v množici zvočnih zapisov, za katere algoritem smatra, da pripadajo enoglasnemu petju.

V primeru zapisa MIDI lahko na trivialen način določimo ali gre za enoglasje ali večglasje. V primeru zvočnega zapisa pa izvedemo algoritem, opisan v nadaljevanju. Vhodni podatek za algoritem za razlikovanje med enoglasnim in večglasnim petjem predstavlja kromagram zvočnega zapisa $k_{n,i}$. Na začetku določimo mejo za vrednosti kromagrama, nad katero upoštevamo, da določen ton pripada melodiji enoglasnega ali večglasnega petja, če pa je moč tona v kromi pod to mejo, se šteje, da ni prisoten ton enoglasja ali večglasja, ampak gre bodisi za harmonik, bodisi za ton, ki je posledica netočnega petja oziroma razmazane krome. Mejo določimo kot tretjino povprečne maksimalne vrednosti krom s formulo 6.4.

$$m = \frac{\sum_{n=0}^{N} max(\{k_{n,i} | i = 0, 1, \ldots, 11\})}{3N} \tag{6.4}$$

V nadaljevanju za posamično kromo določimo, ali pripada večglasju. Če sta v kromi prisotna vsaj dva tona z vrednostjo, večjo od m, ki sta oddaljena za več kot pol tona (sosednja močna tona ne moreta biti posledica večglasja), se šteje, da kroma pripada večglasju. Za določitev zgornjega za posamično kromo k_n uporabimo formulo 6.5.

$$\delta(k_n) = \begin{cases} 1 & max\Big(\{0\} \cup \{min((i - i') \ mod \ 12, (i' - i) \ mod \ 12) \\ & \qquad \big| k_{n,i} > m, k_{n,i'} > m\}\Big) > 1 \\ 0 & \text{sicer} \end{cases} \tag{6.5}$$

Na koncu algoritma še preštejemo število krom izbranega kromagrama, v katerih je prisotno večglasje. Če je število večglasnih krom v kromagramu večje od polovice števila vseh krom, se šteje, da gre pri izbranem zvočnem zapisu za večglasje, sicer pa zvočni zapis pripada enoglasju (formula 6.6).

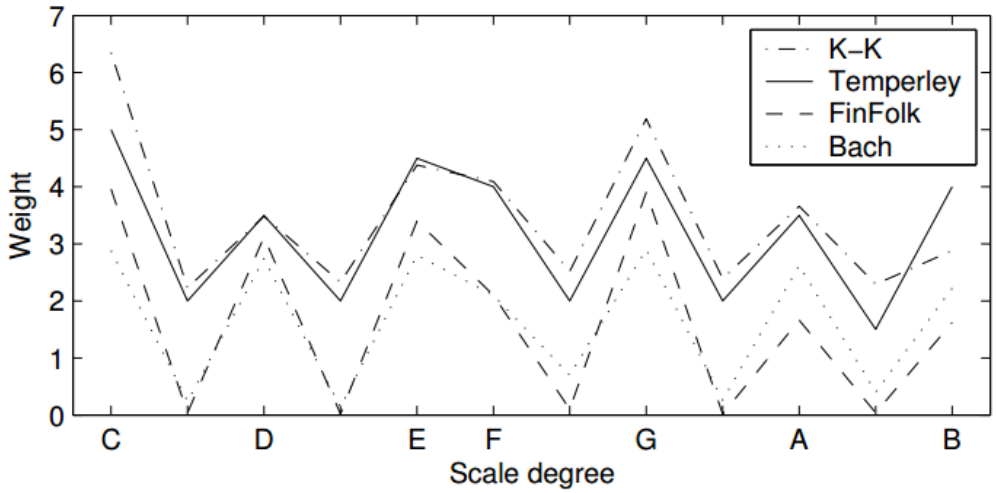
$$VG = \begin{cases} 1 & \sum_{n=0}^{N} \delta(k_n) > \frac{N}{2} \\ 0 & \text{sicer} \end{cases} \tag{6.6}$$

6.3.2 Ostrenje kromagramov s pomočjo verjetnostne porazdelitve tonov v durovi lestvici

Kot smo spoznali s pomočjo analize v poglavju 5, v zvočnih posnetkih pogosto pride do odstopanja velikosti poltona od pravilnega tona neke melodije. Večkrat se nam tudi zgodi, da kroma vsebuje enako močno zastopana sosednja tona. Pri določitvi pravilnega tona si lahko pomagamo z verjetnostno porazdelitvijo zastopanosti tonov v durovi lestvici. Večina slovenskih ljudskih pesmi je namreč zapetih v nekem duru.

Na področju verjetnostnih porazdelitev tonov za različne lestvice predstavlja pomembno delo knjiga [24] avtorice Krumhansl. Za nas je pomembno poznati

porazdelitve tonov lestvic za ljudsko glasbo. Pri tem se opremo na članek [25], v katerem avtorja med drugim analizirata porazdelitve tonov durovih lestvic za arhiv finske ljudske glasbe[26]. Pomembno je dejstvo, da imajo pri ljudski glasbi tako imenovani nelestvični toni mnogo nižjo zastopanost, za razliko od klasične glasbe, ki je analizirana v [24]. Porazdelitve so prikazane na sliki 6.3.



Slika 6.3: Prikaz uteži posameznih tonov za C-dur. Uteži so proporcionalne verjetnosti nastopa tonov v lestvici. S K-K je označena porazdelitev za arhiv klasične glasbe na podlagi članka [24], s Temperley je označena porazdelitev za arhiv klasične glasbe iz članka [27] avtorja Temperleya, s FinFolk je označena porazdelitev za arhiv finske ljudske glasbe. Slika je iz [25].

Na podlagi zgornje analize vidimo, da moramo pri izbiri verjetnostne porazdelitve paziti, da je ta določene na arhivu ljudske glasbe. Izberemo porazdelitev, ki jo v članku [28] predstavi Temperley in je bila izračunana na podlagi essenskega arhiva za ljudsko glasbo [29]. Verjetnostna porazdelitev tonov za essenski arhiv je prikazana v tabeli 6.1.

Iz verjetnostne porazdelitve tonov durove lestvice je tudi zelo jasno razvidna ločnica med toni, ki pripadajo durovi lestvici, in toni, ki ne pripadajo

Stopnja tona	1	#1	2	#2	3	4
Utež	0,184	0,001	0,155	0,003	0,191	0,109
Stopnja tona	#4	5	#5	6	#6	7
Utež	0,005	0,214	0,001	0,078	0,004	0,055

Tabela 6.1: Verjetnostna porazdelitev tonov za skladbe v durovi lestvici. Porazdelitev je določena na podlagi essenskega arhiva za ljudsko glasbo [29].

durovi lestvici. Če pri verjetnostih v tabeli 6.1 za mejno vrednost vzamemo npr. $0,01$, vidimo, da imajo v lestvici zastopani toni 1,2,3,4,5,6,7 verjetnost, večjo od meje, medtem ko imajo toni 1#, 2#, 4#, 5#, 6#, ki niso zastopani v durovi lestvici, verjetnost, manjšo od meje.

Pri našem algoritmu ostrenja najprej predpostavimo, da je večina tonov pravilno zapetih ter na podlagi tega določimo ustrezno durovo lestvico, pripadajočo nekemu posnetku. Durovo lestvico določimo na način, da preizkusimo vseh možnih 12 lestvic. Vseh durovih lestvic je sicer petnajst, vendar gre pri treh parih le za različno poimenovanje istih sestavnih tonov lestvice. Za vsako kandidatno lestvico ustrezno zarotiramo tonski profil p durove lestvice, ki vsebuje vrednosti iz 6.1. Na ta način moramo preizkusiti natanko vseh 12 možnih rotacij tonskega profila durove lestvice. Za vsako rotacijo izračunamo mero podobnosti med histogramom zapisa in zarotiranim tonskim profilom durove lestvice. Za mero podobnosti uporabimo kosinusno razdaljo. Zmaga rotacija k_{min}, pri kateri je razdalja med histogramom zapisa in zarotiranim tonskim profilom durove lestvice minimalna (formula 6.7).

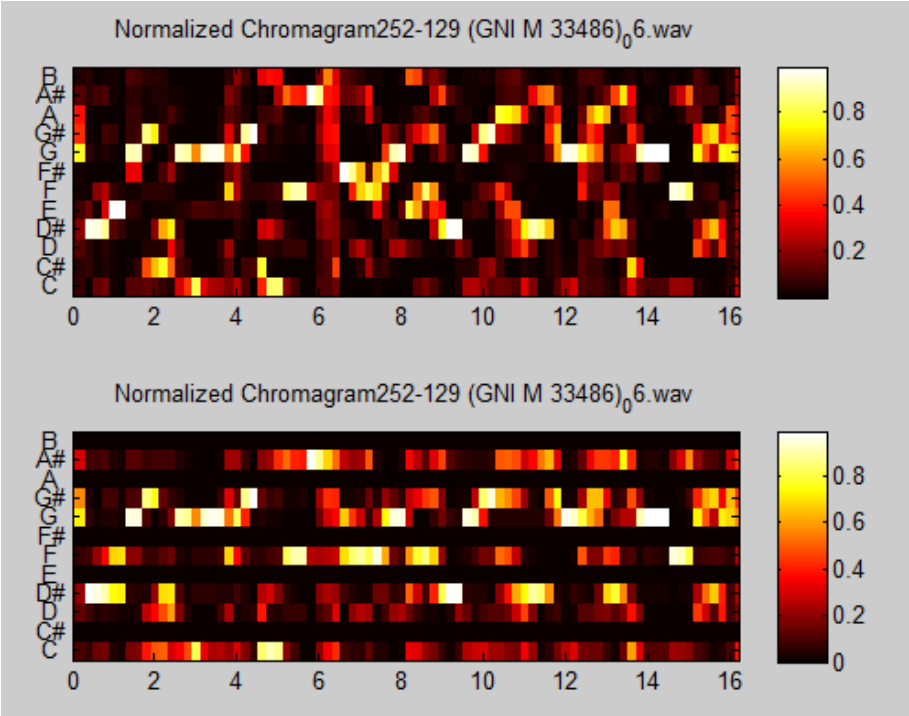
$$\begin{aligned} k_{min} &= argmin_{k=0,1,...,11} d_{cos}(h, R_k(p)) \\ d_{cos}(x,y) &= 1 - \frac{x \cdot y}{\|x\|\|y\|} \\ R_k((p(0),...,p(11)) &= (p((0+k)\ mod\ 12),...,p((11+k)\ mod\ 12)) \end{aligned} \tag{6.7}$$

V nadaljevanju algoritma vrednosti tonov kromagrama, ki niso vsebovani v zmagovalni durovi lestvici, nastavimo na nič, polovično vrednost teh tonov

pa prištejemo obema sosednjima tonoma. Matematično lahko to predstavimo s formulo 6.8. Na ta način dobimo kromagram $\tilde{k}_{n,i}$, ki vsebuje le tone neke durove lestvice.

$$\begin{aligned} g &= (1,0,1,0,1,1,0,1,0,1,0,1) \\ h &= (0,0.5,0,0.5,0,0,0.5,0,0.5,0,0.5,0) \\ \tilde{k}_{n,i} &= R_{k_{min}}(g)(i)k_{n,i} + R_{k_{min}}(h)((i-1)\ mod\ 12)k_{n,(i-1)\ mod\ 12} \\ &\quad + R_{k_{min}}(h)((i+1)\ mod\ 12)k_{n,(i+1)\ mod\ 12} \end{aligned} \tag{6.8}$$

Primer primerjave kromagrama pred in po ostrenju je predstavljen na sliki 6.4.



Slika 6.4: Primerjava kromagramov pred (zgoraj) in po (spodaj) algoritmu ostrenja.

6.3.3 Uporaba različnih mer razdalje med posameznimi kromami pri algoritmu dinamičnega časovnega krivljenja

Pri osnovnem algoritmu za iskanje po arhivu slovenske ljudske glasbe uporabimo za oceno podobnosti dveh zapisov algoritem dinamičnega časovnega krivljenja. V tem algoritmu uporabimo za računanje razdalje med dvema posameznima kromama evklidsko razdaljo 6.9.

$$d(p,q) = \sqrt{\sum_{k=1}^{n}(p_k - q_k)^2} \tag{6.9}$$

Alternativna razdalja, ki jo lahko uporabimo za izboljšanje učinkovitosti algoritma, je kosinusna razdalja med dvema posameznima kromama 6.10.

$$d(p,q) = 1 - \frac{p \cdot q}{\|p\|\|q\|} \tag{6.10}$$

Naslednjo možnost za računanje razdalje med kromami predstavlja razdalja, ki temelji na ideji, predstavljeni v članku J. Serra et al [30]. Pri tej razdalji najprej izvedemo vse možne rotacije ene izmed krom ter za vsako rotacijo izračunamo evklidsko razdaljo med kromama. Nato vzamemo minimalno razdaljo ter preverimo, kateri rotaciji ta minimalna razdalja pripada. V primeru, da pripada rotaciji za 0 mest (alternativno lahko tudi izberemo mesta -1 do +1), določimo, da je razdalja med kromama enaka 0, sicer določimo, da je razdalja med kromama enaka 1 (formula 6.11).

$$d(p,q) = \begin{cases} 0 & \textit{če je rotacija, ki minimizira razdaljo med kromama, enaka 0} \\ 1 & \textit{sicer} \end{cases} \tag{6.11}$$

Alternativno lahko namesto števil 0 in 1 uporabimo tudi druge konstante.

6.3.4 Uporaba uteževanja rezultatov posameznih zapisov, glede na njihovo uspešnost pri ostalih poizvedbah

Naloga našega algoritma je za posamezen glasbeni zapis iz vhodne množice zapisov V najti ustrezen glasbeni zapis v ciljni množici zapisov C. Recimo, da hkrati iščemo ustrezne rezultate za N zapisov iz vhodne množice ter da ciljna množica vsebuje M zapisov. Naš algoritem nam ob tem vrne matriko rezultatov razdalj D med posameznimi zapisi, ki je velikosti N krat M (formula 6.12).

$$D = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{i,1} & d_{i,2} & \cdots & d_{i,m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n,1} & d_{n,2} & \cdots & d_{n,m} \end{bmatrix} \tag{6.12}$$

Posameznemu zapisu $v_i \in V$ iz vhodne množice priredimo tisti zapis $c_j \in C$ iz ciljne množice, ki ima minimalno razdaljo v ustrezni vrstici i, ki pripada v matriki razdalj zapisu v. To lahko zapišemo s formulo 6.13.

$$c_j = min_{c_j \in C} d_{i,j}. \tag{6.13}$$

Izkaže se, da imajo zapisi iz ciljne množice različne povprečne vrednosti razdalj $d_{.j} = (d_{1,j} + d_{2,j} + \ldots + d_{n,j})/N$. To je lahko posledica različne kakovosti petja pri zvočnih zapisih, saj so bolj kakovostno zapeti zvočni posnetki bolj podobni simbolnim zapisom kot manj kakovostno zapeti zvočni posnetki. Tako se lahko zgodi, da je kakovostnejši neustrezen zvočni zapis bolj podoben simbolnemu zapisu kot pripadajoč ustrezen manj kakovosten zvočni zapis. Zato je smiselno utežiti vrstice matrike D na način, da delimo vrstice s povprečno vrednostjo razdalj v vrstici. Na ta način lahko dosežemo večjo točnost algoritma za iskanje po arhivu ljudske glasbe, saj poskrbimo, da lahko manj kakovostno zapeti zvočni posnetki bolj enakovredno tekmujejo z bolj kakovostno zapetimi zvočnimi posnetki pri razvrščanju po podobnosti

simbolnemu zapisu. Uteženo matriko razdalj $\widetilde{D}$ lahko zapišemo s formulo 6.14.

$$\widetilde{D} = \begin{bmatrix} d_{1,1}/d_{.1} & d_{1,2}/d_{.2} & \cdots & d_{1,m}/d_{.m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{i,1}/d_{.1} & d_{i,2}/d_{.2} & \cdots & d_{i,m}/d_{.m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n,1}/d_{.1} & d_{n,2}/d_{.2} & \cdots & d_{n,m}/d_{.m} \end{bmatrix} \tag{6.14}$$

Poglavje 7

Ovrednotenje uspešnosti algoritmov

Uspešnost algoritmov lahko ovrednotimo na različnih vrstah vhodnih in ciljnih glasbenih zapisov. V nadaljevanju se bomo osredotočili na tri osnovne scenarije:

1. Za simbolične zapise glasbe iščemo ustrezne zvočne zapise.
2. Za zvočne zapise iščemo ustrezne simbolične zapise glasbe.
3. Za zvočne zapise iščemo ustrezne zvočne zapise.

Za ustrezno vrednotenje je najprej potrebno ustrezno označiti zvočne in simbolične zapise glasbe. Tako med drugim določimo, kateri zvočni in simbolični zapisi so si med seboj sorodni, kar pomeni, da pripadajo neki skupni melodiji ljudske pesmi. Pri tem ne gre le za to, da določimo ustrezne pare zvočnih in simboličnih zapisov, ampak lahko npr. več različnih zvočnih posnetkov pripada isti melodiji ljudske pesmi. Označitev izvedemo na podlagi poslušanja posnetkov ter primerjav posameznih kromagramov. Glede na to označitev ločimo dva načina evalvacije:

a) Za ustrezen zadetek štejemo, kadar algoritem za določen simboličen ali zvočni zapis vrne točno določen pripadajoč zvočni zapis (če je vhodni

zapis v simbolni obliki) ali simbolni zapis (če je vhod zvočni zapis). Ta način evalvacije uporabimo pri scenarijih 1 in 2.

b) Za ustrezen zadetek štejemo, kadar algoritem za določen simboličen ali zvočni zapis vrne soroden simboličen ali zvočni zapis glede na našo predhodno označitev zapisov. Omenjen način evalvacije uporabimo pri scenariju 3.

Pri vrednotenju algoritmov uporabimo podobne mere, kot so v uporabi pri vsakoletnem izzivu MIREX [31], bolj natančno pri nalogi poizvedba na podlagi petja, kjer za določen posnetek petja iščejo ustrezen simbolni zapis. Uporabimo naslednje mere:

m1) Odstotek poizvedb, kjer se ustrezen pripadajoč zapis znajde na prvem mestu.

m2) Odstotek poizvedb, kjer se ustrezen pripadajoč zapis znajde med prvimi tremi vrnjenimi zapisi.

m3) Odstotek poizvedb, kjer se ustrezen pripadajoč zapis znajde med prvimi desetimi vrnjenimi zapisi.

m4) Povprečno mesto prvega ustreznega vrnjenega zapisa.

7.1 Rezultati

Najprej analiziramo vpliv frekvence vzorčenja kromagramov na uspešnost rezultatov osnovnega algoritma brez izboljšav. Analizo opravimo za scenarij 1. Dobimo tabelo rezultatov 7.1.

Vidimo, da dosežemo najboljše rezultate pri frekvenci vzorčenja 5 oziroma 10 vzorcev na sekundo. Ker manjše število vzorcev pomeni hitrejšo izvedbo algoritmov, v nadaljevanju izberemo frekvenco vzorčenja 5 vzorcev na sekundo za standardno frekvenco vzorčenja in preizkusimo naše izboljšave osnovnega algoritma.

Frekvenca(1/s)	1	3	5	10
m3 (%)	42,6	47,9	51,0	51,0

Tabela 7.1: Uspešnost osnovnega algoritma v odvisnosti od frekvence vzorčenja kromagramov.

7.1.1 Iskanje ustreznega zvočnega zapisa za posamezen zapis MIDI

Najprej se osredotočimo na scenarij, ko za posamezen zapis MIDI iščemo ustrezen zvočni zapis. Baza zvočnih zapisov vsebuje 200 posnetkov, iskanje pa izvedemo za 94 različnih zapisov MIDI. Rezultati za osnovni algoritem so predstavljeni v tabeli 7.2.

Algoritem	m1 (%)	m2 (%)	m3 (%)	m4
Osnovni algoritem	26,6	37,2	51,0	23,1

Tabela 7.2: Uspešnost osnovnega algoritma za iskanje ustreznega zvočnega zapisa za posamezen zapis MIDI.

Vidimo, da se med prvimi desetimi vrnjenimi rezultati le pri 51,0 % poizvedb nahaja ustrezen zvočni posnetek. Sedaj preizkusimo naše izboljšave, ki smo jih predstavili v poglavju 6.3. Rezultati posameznih izboljšav so prikazani v tabeli 7.3.

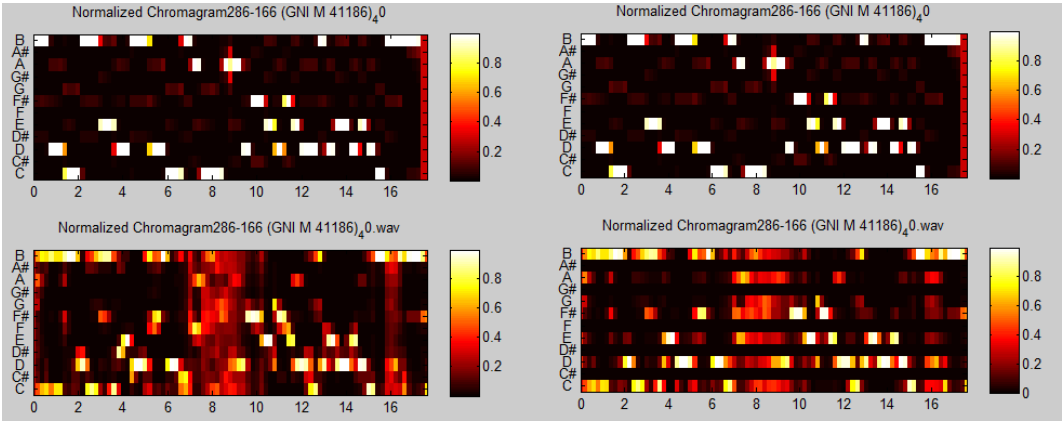
Iz tabele rezultatov je razvidno, da posamične izboljšave, kot so algoritem ostrenja kromagramov s pomočjo verjetnostne porazdelitve tonov v durovi lestvici, algoritem z razdaljo Serra, algoritem s kosinusno razdaljo in algoritem uteževanja rezultatov, dosežejo izboljšanje rezultatov osnovnega algoritma. Izmed vseh posamičnih izboljšav se največje izboljšanje rezultatov doseže z uporabo kosinusne razdalje. V spodnjem delu tabele 7.3 so prikazani rezultati za različne kombinacije izboljšav. Najprej vidimo, da algoritem ostrenja kromagramov z uporabo kosinusne razdalje doseže napredek pri meri m1 (z 36,2 % na 40,4 %) v primerjavi z algoritmom, ki uporabi le kosinusno razda-

Algoritem	m1 (%)	m2 (%)	m3 (%)	m4
Osnovni algoritem	26,6	37,2	51,0	23,1
Algoritem ostrenja kromagramov za dur in algoritem razlikovanja enoglasja ter večglasja	31,9	44,7	52,1	31,6
Algoritem z razdaljo Serra	26,6	42,6	61,7	26,9
Algoritem s kosinusno razdaljo	36,2	52,1	62,8	18,7
Algoritem uteževanja rezultatov	36,2	46,8	58,5	14,8
Algoritem ostrenja kromagramov za dur in algoritem razlikovanja enoglasja ter večglasja in kosinusna razdalja	40,4	50,0	59,5	28,6
Algoritem ostrenja kromagramov za dur in algoritem razlikovanja enoglasja ter večglasja in uteževanje rezultatov	34,0	44,7	55,3	30,0
Algoritem z razdaljo Serra in uteževanje rezultatov	23,4	36,2	55,3	33,0
Algoritem s kosinusno razdaljo in uteževanje rezultatov	44,7	55,3	74,5	12,5
Algoritem ostrenja kromagramov za dur in algoritem razlikovanja enoglasja ter večglasja in kosinusna razdalja in uteževanje rezultatov	41,5	52,1	62,8	30,4

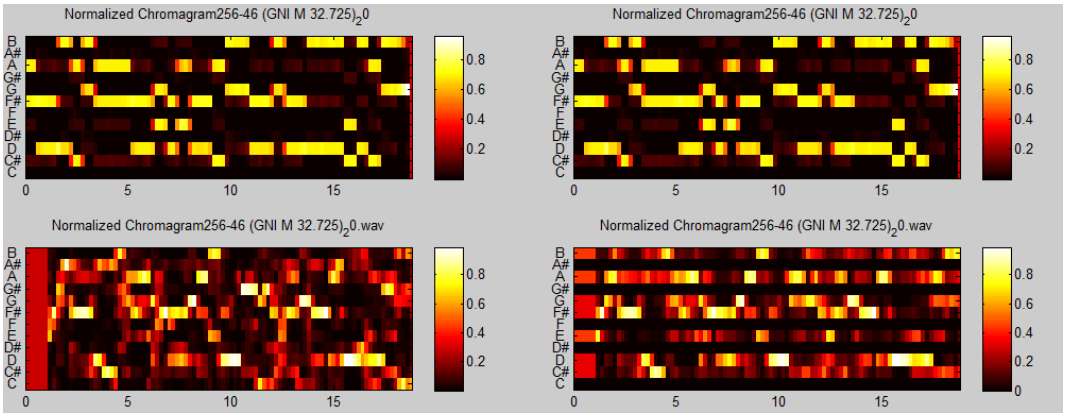
Tabela 7.3: Uspešnost različnih izboljšav osnovnega algoritma za iskanje ustreznega zvočnega zapisa za posamezen zapis MIDI.

ljo. Manj uspešna pa je omenjena kombinacija pri preostalih merah m2 do m4. Algoritem ostrenja kromagramov doseže boljše rezultate na bolj kvalitetnih posnetkih, kot je prikazano na sliki 7.1. Prav tako ostrenje večkrat pomaga v primeru dvoglasja, kot je prikazano na sliki 7.2. Tudi v tem primeru imamo opravka s precej dobro izvedbo petja. Kadar pa je petje slabše

kvalitete, lahko ostrenje poslabša rezultat, kot to velja za primer na sliki 7.3.

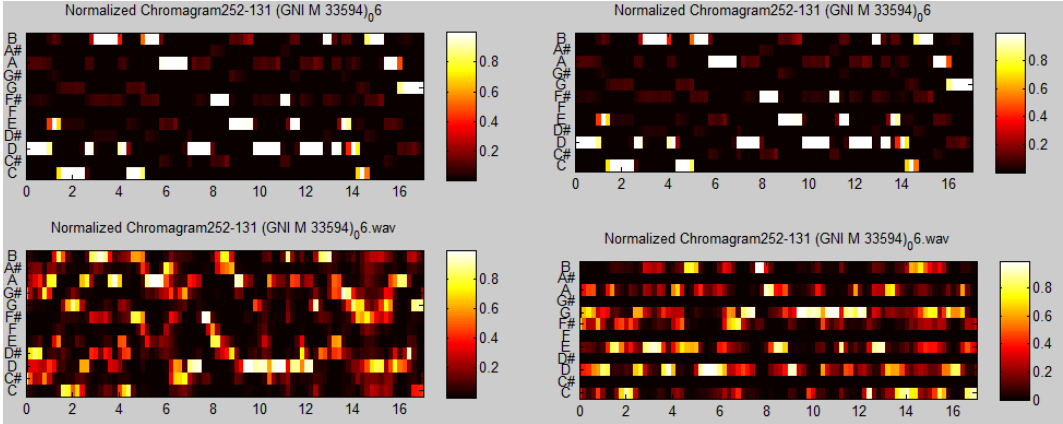


Slika 7.1: Primerjava kromagrama brez ostrenja (levo spodaj) s kromagramom z ostrenjem (desno spodaj). Kromagram brez ostrenja se pri algoritmu iskanja uvrsti na drugo mesto, kromagram z ostrenjem pa na prvo mesto.



Slika 7.2: Primerjava kromagrama brez ostrenja (levo spodaj) s kromagramom z ostrenjem (desno spodaj) za primer dvoglasnega petja. Kromagram z ostrenjem se pri algoritmu iskanja uvrsti precej višje, kot kromagram brez ostrenja.

V nadaljevanju tabele so prikazani rezultati kombiniranja izboljšav z uteževanjem rezultatov. Vidimo, da uteževanje rezultatov doseže izboljšanje



Slika 7.3: Primerjava kromagrama brez ostrenja (levo spodaj) s kromagramom z ostrenjem (desno spodaj). Kromagram brez ostrenja se pri algoritmu iskanja uvrsti precej višje, kot kromagram z ostrenjem. To je posledica slabše kvalitete petja. V konkretnem primeru pride v poteku algoritma tudi do napačne določitve skupnega ključa zapisa MIDI in zvočnega posnetka.

uspešnosti algoritmov v večini primerov, izjema je razdalja, ki jo je predlagal Serra. Najboljši rezultat izmed vseh algoritmov doseže uporaba kosinusne razdalje v kombinaciji z algoritmom za uteževanje rezultatov. Če k omenjeni kombinaciji dodamo še algoritem ostrenja kromagramov ne dosežemo nadaljnjega izboljšanja rezultatov pri meri m1, kot smo ga dosegli z dodatkom ostrenja k algoritmu s kosinusno razdaljo. To je posledica slabših rezultatov algoritma ostrenja za slabše zapete posnetke (uteževanje doseže enakovredno obravnavo posnetkov glede na kvaliteto petja).

7.1.2 Iskanje ustreznega zapisa MIDI za posamezen zvočni zapis

V nadaljevanju se osredotočimo na scenarij, ko za posamezen zvočni zapis iščemo ustrezen zapis MIDI. Baza zapisov MIDI vsebuje 200 zapisov, iskanje pa izvedemo za 94 različnih zvočnih zapisov. Rezultati so prikazani v tabeli 7.4.

Vidimo, da tokrat pri osnovnem algoritmu za razliko od iskanja zvočnega zapisa za zapis MIDI dosežemo zelo nizko število pravilnih poizvedb na prvem mestu (3,2 %). Uspešnost za mero m3 pa je podobna. Vse posamične izboljšave osnovnega algoritma dosežejo izboljšanje rezultatov. Iz rezultatov je razvidno, da pri uspešnosti posamičnih izboljšav tokrat najbolj izstopa uteževanje rezultatov, kar je verjetno posledica tega, da so rezultati algoritma DTW močno odvisni od samega zapisa MIDI (pri algoritmu DTW iščemo podzaporedje zvočnega zapisa, ki se najbolj prilega zapisu MIDI). Podobno kot pri iskanju ustreznega zvočnega posnetka za zapis MIDI, se tudi tokrat v celoti najbolj obnese algoritem s kosinusno razdaljo in uteževanjem rezultatov. Ta algoritem prepričljivo zmaga tudi pri meri m3, kjer doseže 80,9% natančnost. Če opazujemo le mero m1, pa se tokrat najbolje obnese algoritem ostrenja kromagramov z uporabo kosinusne razdalje pri algoritmu DTW in uteževanjem končne matrike rezultatov, ki doseže 43,6 odstotno natančnost. Ostrenje torej izboljša rezultate algoritma s kosinusno razdaljo in uteževanjem za mero m1. Pri iskanju zvočnega zapisa za simbolični zapis to ni bilo res, saj smo tedaj uteževali različno kakovostne zvočne zapise, ostrenje pa je manj zanesljivo za slabše zvočne posnetke. Tokrat utežujemo simbolične zapise, zato prej omenjenih težav nimamo.

Algoritem	m1 (%)	m2 (%)	m3 (%)	m4
Osnovni algoritem	3,2	25,5	47,9	25,8
Algoritem ostrenja kromagramov za dur in algoritem razlikovanja enoglasja ter večglasja	1,0	26,6	51,1	36,2
Algoritem z razdaljo Serra	10,6	25,5	45,7	46,0
Algoritem s kosinusno razdaljo	9,6	35,1	62,8	19,6
Algoritem uteževanja rezultatov	28,7	46,8	70,2	12,1
Algoritem ostrenja kromagramov za dur in algoritem razlikovanja enoglasja ter večglasja in kosinusna razdalja	10,6	40,4	60,6	34,8
Algoritem ostrenja kromagramov za dur in algoritem razlikovanja enoglasja ter večglasja in uteževanje rezultatov	28,7	46,8	62,8	27,7
Algoritem z razdaljo Serra in uteževanje rezultatov	31,9	40,4	56,4	35,8
Algoritem s kosinusno razdaljo in uteževanje rezultatov	38,3	64,9	80,9	11,4
Algoritem ostrenja kromagramov za dur in algoritem razlikovanja enoglasja ter večglasja in kosinusna razdalja in uteževanje rezultatov	43,6	59,6	67,0	29,2

Tabela 7.4: Uspešnost različnih izboljšav osnovnega algoritma za iskanje ustreznega zapisa MIDI za posamezen zvočni zapis.

7.1.3 Iskanje ustreznega sorodnega zvočnega zapisa za posamezen zvočni zapis

V nadaljevanju se osredotočimo na scenarij, ko za posamezen zvočni zapis iščemo ustrezen soroden zvočni zapis. Baza zvočnih zapisov vsebuje 200

posnetkov, iskanje pa izvedemo za 24 različnih zvočnih zapisov. Rezultati so predstavljeni v tabeli 7.5.

Algoritem	m1 (%)	m2 (%)	m3 (%)	m4
Osnovni algoritem	45,8	50,0	58,3	32,8
Algoritem ostrenja kromagramov za dur in algoritem razlikovanja enoglasja ter večglasja	29,2	37,5	41,7	28,6
Algoritem z razdaljo Serra	33,3	45,8	62,5	30,7
Algoritem s kosinusno razdaljo	45,8	54,2	62,5	36,9
Algoritem uteževanja rezultatov	20,8	41,7	58,3	42,7
Algoritem ostrenja kromagramov za dur in algoritem razlikovanja enoglasja ter večglasja in kosinusna razdalja	45,8	50,0	58,3	37,4
Algoritem ostrenja kromagramov za dur in algoritem razlikovanja enoglasja ter večglasja in uteževanje rezultatov	33,3	33,3	37,5	74,8
Algoritem z razdaljo Serra in uteževanje rezultatov	25,0	25,0	33,3	51,1
Algoritem s kosinusno razdaljo in uteževanje rezultatov	33,3	41,7	54,2	39,9
Algoritem ostrenja kromagramov za dur in algoritem razlikovanja enoglasja ter večglasja in kosinusna razdalja in uteževanje rezultatov	29,2	33,3	41,7	58,6

Tabela 7.5: Uspešnost različnih izboljšav osnovnega algoritma za iskanje ustreznega sorodnega zvočnega zapisa za posamezen zvočni zapis.

Iz tabele rezultatov je razvidno, da osnovni algoritem doseže boljše rezultate kot pri ostalih dveh scenarijih iskanj, kar pa je posledica različnih testnih množic, saj tokrat iskanje izvedemo le za 24 različnih zvočnih zapisov, za ka-

tere imamo na voljo sorodne zvočne zapise. Iz tabele rezultatov je razvidno tudi, da se algoritem uteževanja rezultatov ne obnese dobro, saj večinoma poslabša uspešnost. To je verjetno posledica manjše množice poizvedb (24), na kateri določimo uteži za zvočne zapise. Najboljše rezultate za meri m2 in m3 doseže algoritem s kosinusno razdaljo, za mero m4 pa je najuspešnejši algoritem ostrenja kromagramov z uporabo verjetnostne porazdelitve durovih lestvic.

Poglavje 8

Sklepne ugotovitve

V magistrskem delu smo najprej izvedli pregled stanja na področju poravnave zvočnih in notnih zapisov ter iskanja po glasbenih arhivih. Za cilj smo si zadali prilagoditi omenjene metode za uporabo na arhivih ljudske glasbe. S tem namenom najprej analiziramo značilnosti ljudske glasbe ter značilnosti posnetkov petja ljudskih pevcev. Na podlagi analize algoritmov za iskanje po arhivih ljudske glasbe smo sprejeli odločitev, da za samo poravnavo zapisov izberemo algoritem dinamičnega časovnega krivljenja. V nadaljevanju dela predlagamo nekaj izboljšav standardnega postopka za iskanje po arhivu ljudske glasbe. Ker se pri ljudski glasbi srečujemo z netočnostjo petja in drsenjem intonacije, v magistrskem delu predlagamo metodo za ostrenje kromagramov. Ta metoda izkoristi dejstvo, da je porazdelitev tonov pri ljudski glasbi mnogo bolj določena, kot to velja za klasično glasbo. Z določitvijo ustrezne lestvice za posamezno ljudsko pesem lahko nekatere neustrezno zapete tone premaknemo na njihovo bolj verjetno mesto v kromagramu. V magistrskem delu predstavimo tudi različne mere za računanje podobnosti med kromami, kot so evklidska razdalja, kosinusna razdalja ter razdalja, ki jo je v članku [30] predlagal Serra. Nadalje v magistrskem delu predstavimo algoritem za uteževanje rezultatov posameznih zapisov glede na njihovo uspešnost pri preostalih poizvedbah.

Vse algoritme preizkusimo na arhivu slovenske ljudske glasbe. Pri testiranju ločimo tri scenarije, in sicer iskanje zvočnega posnetka na podlagi zapisa MIDI, iskanje zapisa MIDI na podlagi zvočnega posnetka ter iskanje sorodnega zvočnega posnetka na podlagi zvočnega posnetka. Za vrednotenje algoritmov uporabimo različne mere. Večina predlaganih izboljšav doseže napredek v primerjavi s standardnim algoritmom. Izkaže se, da nasplošno najboljše rezultate doseže kombinacija uporabe kosinusne razdalje za računanje razdalj med kromami in metoda končnega uteževanja rezultatov. Metoda ostrenja kromagramov dosega zelo dobre rezultate predvsem pri meri odstotka pravilnih zadetkov na prvem mestu vrnjenih rezultatov.

Predstavljena metoda dosega dobro natančnost pri iskanje po arhivu ljudske glasbe, ki vsebuje nekaj sto posnetkov. V uvodu smo navedli, da se metode, ki temeljijo na algoritmu DTW večinoma uporabljajo v zadnjem delu sistemov QBH zaradi relativne časovne zahtevnosti algoritma DTW. Pri velikosti našega arhiva to ne predstavlja težav, saj se celotna procedura iskanja za posamičen zvočni zapis izvede v nekaj sekundah. V nadaljnjem delu, bi lahko na podlagi značilnosti ljudske glasbe poskusili raziskati tudi možne prilagoditve algoritmov, ki se uporabljajo v začetnem delu sistemov QBH in na ta način naš algoritem razširili tudi za mnogo večje glasbene arhive.

V magistrskem delu smo videli, da si lahko pri iskanju pomagamo s specifičnimi značilnostmi glasbenega arhiva, v našem primeru smo izkoristili dejstvo, da ima večina ljudskih pesmi, za razliko od klasične glasbe, precej določeno verjetnostno porazdelitev posameznih tonov. To lastnost glasbe bi lahko poskusili uporabiti tudi pri iskanju na arhivih drugih zvrsti glasbe, kot je na primer pop glasba.

Literatura

[1] M. Miron, J. J. Carabias, J. Janer, Improving score-informed source separation for classical music through note refinement, in: 16th International Society for Music Information Retrieval (ISMIR) Conference, Malaga, 2015.

[2] J. J. Carabias, Musical instrument models estimation for polyphonic music transcription, Ph.D. thesis, University of Jaen, Linares (Spain) (12/2011 2011).

[3] S. Salvador, P. Chan, Toward accurate dynamic time warping in linear time and space, Intell. Data Anal. 11 (5) (2007) 561–580.
URL `http://dl.acm.org/citation.cfm?id=1367985.1367993`

[4] Z. Ghahramani, Hidden markov models, World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2002, Ch. An Introduction to Hidden Markov Models and Bayesian Networks, pp. 9–42.
URL `http://dl.acm.org/citation.cfm?id=505741.505743`

[5] A. Cont, A coupled duration-focused architecture for real-time music-to-score alignment, in: Pattern Analysis and Machine Intelligence, IEEE Transactions, 2009, pp. 974–987.

[6] N. Montecchio, A. Cont, A unified approach to real time audio-to-score and audio-to-audio alignment using sequential monte carlo inference techniques, in: Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference, 2011, pp. 193–196.

[7] J. Carabias-Orti, F. Rodriguez-Serrano, P. Vera-Candeas, N. Ruiz-Reyes, F. Canadas-Quesada, An audio to score alignment framework using spectral factorization and dynamic time warping, in: 16th International Society for Music Information Retrieval (ISMIR) Conference, 2015., 2015.
URL http://ismir2015.uma.es/articles/94_Paper.pdf

[8] M. Miron, J. J. Carabias, J. Janer, Audio-to-score alignment at the note level for orchestral recordings, in: 15th International Society for Music Information Retrieval Conference, 2014.

[9] A. Ghias, J. Logan, D. Chamberlin, B. C. Smith, Query by humming: musical information retrieval in an audio database, in: Proceedings of the third ACM international conference on Multimedia, ACM, 1995, pp. 231–236.

[10] R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, R. Van Oostrum, et al., Using transportation distances for measuring melodic similarity., in: ISMIR, 2003.

[11] J.-S. R. Jang, H.-R. Lee, A general framework of progressive filtering and its application to query by singing/humming, Audio, Speech, and Language Processing, IEEE Transactions on 16 (2) (2008) 350–358.

[12] J.-S. R. Jang, C.-L. Hsu, H.-R. Lee, Continuous hmm and its enhancement for singing/humming query retrieval., in: ISMIR, Citeseer, 2005, pp. 546–551.

[13] R. B. Dannenberg, W. P. Birmingham, G. Tzanetakis, C. Meek, N. Hu, B. Pardo, The musart testbed for query-by-humming evaluation, Computer Music Journal 28 (2) (2004) 34–48.

[14] M. Müller, P. Grosche, F. Wiering, Towards automated processing of folk song recordings, in: E. Selfridge-Field, F. Wiering, G. A. Wiggins (Eds.), Knowledge representation for intelligent music processing, no.

09051 in Dagstuhl Seminar Proceedings, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, Dagstuhl, Germany, 2009.
URL http://drops.dagstuhl.de/opus/volltexte/2009/1966

[15] M. Müller, P. Grosche, F. Wiering, Automated analysis of performance variations in folk song recordings, in: Proceedings of the International Conference on Multimedia Information Retrieval (MIR), Philadelphia, Pennsylvania, USA, 2010, pp. 247–256.

[16] Y. Suzuki, V. Mellert, U. Richter, H. Møller, L. Nielsen, R. Hellman, K. Ashihara, K. Ozawa, H. Takeshima, Precise and full-range determination of two-dimensional equal loudness contours, Tohoku University, Japan.

[17] M. Russ, Sound Synthesis and Sampling, 1st Edition, Butterworth-Heinemann, Newton, MA, USA, 1996.

[18] P. Roland, Xml4mir: Extensible markup language for music information retrieval, in: Proceedings of the 1st International Conference on Music Information Retrieval, Plymouth (Massachusetts), USA, 2000, http://ismir2000.ismir.net/papers/roland_paper.pdf.

[19] MIDI Manufacturers Association, The complete midi 1.0 detailed specification (1996).

[20] M. Müller, Information Retrieval for Music and Motion, Springer, 2007.
URL https://books.google.si/books?id=kSzeZWR2yDsC

[21] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286. doi:10.1109/5.18626.
URL http://dx.doi.org/10.1109/5.18626

[22] Michalis Morakeas, Audio to Score Alignment using Hidden Markov Models (2014).

URL http://nefeli.lib.teicrete.gr/browse/sefe/mta/2014/MorakeasMichail/attached-document-1402913270-993820-10855/MorakeasMichail2014.pdf

[23] G. Donaj, Algoritmi za reševanje treh osnovnih problemov prikritih markovskih modelov, diplomsko delo, Univerza v Mariboru, Fakulteta za naravoslovje in matematiko (2011).

[24] C. L. Krumhansl, Cognitive foundations of musical pitch, Oxford University Press, 2001.

[25] S. T. Madsen, G. Widmer, J. Kepler, Key-finding with interval profiles, in: Proceedings of the International Computer Music Conference (ICMC), 2007, pp. 27–31.

[26] Tuomas Eerola, Petri Toiviainen, Suomen Kansan eSavelmat. Finnish Folk Song Database. (2004).
URL http://www.jyu.fi/musica/sks/

[27] D. Temperley, What's key for key? the krumhansl-schmuckler key-finding algorithm reconsidered, Music Perception: An Interdisciplinary Journal 17 (1) (1999) 65–100.

[28] D. Temperley, E. W. Marvin, Pitch-class distribution and the identification of key, Music Perception: An Interdisciplinary Journal 25 (3) (2008) 193–212.

[29] H. Schaffrath, The Essen Folksong Collection in Kern Format. (1995).

[30] J. Serrà, E. Gómez, P. Herrera, X. Serra, Chroma binary similarity and local alignment applied to cover song identification, IEEE Transactions on Audio, Speech and Language Processing 16 (2008) 1138–1151. doi:10.1109/TASL.2008.924595.
URL files/publications/jserraTSALP08.pdf

[31] J. Downie, K. West, A. Ehmann, E. Vincent, The 2005 music information retrieval evaluation exchange (mirex 2005): Preliminary overview, in: 6th Int. Conf. on Music Information Retrieval (ISMIR), 2005, pp. 320–323.